Learning to Optimize*

George W. Evans          Bruce McGough
University of Oregon      University of Oregon
University of St. Andrews

May 16, 2014

Abstract

We consider decision-making by boundedly-rational agents in dynamic stochastic environments. The behavioral primitive is anchored to the shadow price of the state vector. Our agent forecasts the value of an additional unit of the state tomorrow using estimated models of shadow prices and transition dynamics, and uses this forecast to choose her control today. The control decision, together with the agent’s forecast of tomorrow’s shadow price, are then used to update the perceived shadow price of today’s states. By following this boundedly-optimal procedure the agent’s decision rule converges over time to the optimal policy. Specifically, within standard linear-quadratic environments, we obtain general conditions for asymptotically optimal decision-making: agents learn to optimize. Our results carry over to closely related procedures based on value-function learning and Euler-equation learning. We provide examples showing that shadow-price learning extends to general dynamic-stochastic decision-making environments and embeds naturally in general-equilibrium models.

JEL Classifications: E52; E31; D83; D84

Key Words: Learning, Optimization, Bellman Systems

*We thank Richard Dennis and Eric Leeper for helpful suggestions, and are indebted for the many comments from participants at numerous seminars and conferences including The University of Oregon, The University of St. Andrews, FRB St. Louis Learning Week workshop, Indiana University-Bloomington, The Bank of Spain, Banque du France, University of California-Irvine, University of Pittsburg, West Virginia University. Financial support from National Science Foundation Grant no. SES-1025011 is gratefully acknowledged.
**1 Introduction**

A central paradigm of modern macroeconomics is the need for micro-foundations. Macroeconomists construct their models by aggregating the behavior of individual agents who are assumed “rational” in two important ways: they form forecasts optimally; and, given these forecasts, they make choices by maximizing their objective. Together with simple market structures, and sometimes institutional frictions, it is this notion of rationality that identifies a micro-founded model. While assuming rationality is at the heart of much economic theory, the implicit sophistication required of agents in the benchmark “rational expectations” equilibrium,\(^1\) both as forecasters and as decision theorists, is substantial: they must be able to form expectations conditional on the true distributions of the endogenous variables in the economy; and they must be able to make choices – i.e. solve infinite horizon programming problems – given these expectations.

The criticism that the ability to make optimal forecasts requires an unrealistic level of sophistication has been leveled repeatedly; and, in response to this criticism, a literature on bounded rationality and adaptive learning has developed. Boundedly rational agents are not assumed to know the true distributions of the endogenous variables; instead, they have forecasting models that they use to form expectations. These agents update their forecasting models as new data become available, and through this updating process the dynamics of the associated economy can be explored. In particular, the asymptotic behavior of the economy can be analyzed, and if the economy converges in some natural sense to a rational expectations equilibrium, then we may conclude that agents in the economy are able to learn to forecast optimally.

In this way, the learning literature has provided a response to the criticism that rational expectations is unrealistic. Early work on least-squares (and, more generally, adaptive) learning in macroeconomics includes Bray (1982), Bray and Savin (1986) and Marcet and Sargent (1989); for a systematic treatment, see Evans and Honkapohja (2001). Convergence to rational expectations is not automatic and “expectational stability” conditions can be computed to determine local stability. Recent applications have emphasized the possibility of novel learning dynamics that may also arise in some models.

Increasingly, the adaptive learning approach has been applied to dynamic stochastic general equilibrium (DSGE) models by incorporating learning into a system of expectational difference equations obtained from linearizing conditions that capture optimizing behavior and market equilibrium. We will discuss this procedure later, but for now we emphasize that because the representative agents in these models typically live forever, they are being assumed to be optimal decision makers,

\(^1\)Seminal papers of the rational expectations approach include, e.g., Muth (1961), Lucas (1972) and Sargent (1973).
solving difficult stochastic dynamic optimization problems, despite having bounded rationality as forecasters. We find this discontinuity in sophistication unsatisfactory as a model of individual agent decision-making. The difficulty that subjects have in making optimal decisions, given their forecasts, has lead experimental researchers to distinguish between “learning to forecast” and “learning to optimize” experiments.\textsuperscript{2} For example, in recent experimental work, Bao, Duffy, and Hommes (2013) find that in a cobweb setting making optimal decisions is as difficult as making optimal forecasts.

To address this discontinuity we define the notion of \textit{bounded optimality}. We imagine our agents facing a sequence of decision problems in an uncertain environment: not only is there uncertainty in that the environment is inherently stochastic, but also our agents do not fully understand the conditional distributions of the variables requiring forecasts. One option when modeling agent decisions in this type of environment is to assume that agents are Bayesian and that, given their priors, they are able to fully solve their dynamic programming problems. However, we feel this level of sophistication is extreme, and instead, we prefer to model our agents as relying on decidedly simpler behavior. Informally, we assume that each day our agents act as if they face a two-period optimization problem: they think of the first period as “today” and the second period as “the future,” and use one-period-ahead forecasts of shadow prices to measure the trade-off between choices today and the impact of these choices on the future. We call our implementation of bounded optimality \textit{shadow price learning} (SP-learning).

Our notion of bounded optimality is inexorably linked to bounded rationality: agents in our economy are not assumed to fully understand the conditional distributions of the economy’s variables, or, in the context of an individual’s optimization problem, the conditional distributions of the state variables. Instead, consistent with the adaptive learning literature, we provide our agents with forecasting models, which they re-estimate as new data become available. Our agents use these estimated models to make one-period forecasts, and then use these one-period forecasts to make decisions.

We find our learning mechanism appealing for a number of reasons: it requires only simple econometric modeling and thus is consistent with the learning literature; it assumes agents make only one-period-ahead forecasts instead of establishing priors over the distributions of all future endogenous variables; and it imposes only that agents make decisions based on these one-period-ahead forecasts, rather than requiring agents to solve a dynamic programming problem with parameter uncertainty. Finally, SP-learning postulates that, fundamentally, agents make decisions by facing suitable prices for their trade-offs. This is a hallmark of economics. The central question that we address is whether SP-learning can converge asymptotically

\textsuperscript{2}This issue is discussed in Marimon and Sunder (1993), Marimon and Sunder (1994) and Hommes (2011). The distinction was also noted in Sargent (1993).
to fully optimal decision making. This is the analog of the original question, posed in the adaptive learning literature, of whether least-squares learning can converge asymptotically to rational expectations. Our main result is that convergence to fully optimal decision-making can indeed be demonstrated in the context of the standard linear-quadratic setting for dynamic decision-making.

Although we focus on SP-learning, we also consider two alternative implementations of bounded optimality: value-function learning and Euler-equation learning. Under value-function learning agents estimate and update (a model) of the value function, and make decisions based on the implied shadow prices given by the derivative of the estimated value function. With Euler-equation learning agents bypass the value-function entirely and instead make decisions based on an estimated model of their own policy rule. We establish that our central convergence results extend to these alternative implementations.

Our paper is organized as follows. In Section 2 we provide an overview of alternative approaches and introduce our technique. In Section 3 we investigate the regulator’s problem in a standard linear-quadratic framework. We show, under quite general conditions, that the policy rule employed by our boundedly optimal regulator converges to the optimal policy rule: following our simple behavioral primitives, our regulator learns to optimize. This is our central theoretical result, given as Theorem 4 in Section 3. Section 3 also provides a general comparison of SP-learning with alternative implementations, including value-function learning and Euler-equation learning. We note that while there are many applications of Euler-equation learning in the literature, our Theorem 7 is the first to establish its asymptotic optimality at the agent level in a general setting. Theorem 6 establishes the corresponding result for value-function learning. Section 4 introduces shadow-price learning into a single agent problem in a simple economic setting; we characterize individual agents’ decisions as based on their boundedly rational forecasts, and analyze associated learning dynamics. Section 5 illustrates SP-learning within a Ramsey model. Section 6 concludes.

2 Background and Motivation

Before turning to a systematic presentation of our results we first, in this Section, review the most closely related approaches available in the literature, and we then introduce and motivate our general methodology and discuss how it relates to the existing literature.
2.1 Agent-level learning and decision-making

We are, of course, not the first to address the issues outlined in the Introduction. A variety of agent-level learning and decision-making mechanisms, differing both in imposed sophistication and conditioning information, have been advanced. Here we briefly summarize these contributions, beginning with those that make the smallest departure from the benchmark rational expectations hypothesis.

Cogley and Sargent (2008) consider Bayesian decision making in a permanent-income model with risk aversion. In their set-up, income follows a two-state Markov process with unknown transition probabilities, which implies that standard dynamic programming techniques are not immediately applicable. A traditional bounded rationality approach is to embrace Kreps' “anticipated utility” model, in which agents determine their program given their current estimates of the unknown parameters. Instead, Cogley and Sargent (2008) treat their agents as Bayesian econometricians, who use recursively updated sufficient statistics as part of an expanded state space to specify their programming problem’s time-invariant transition law. In this way agents are able to compute the fully optimal decision rule. The authors find that the fully optimal solution in their set-up is only a marginal improvement on the boundedly optimal procedure of Kreps. This is particularly interesting because to obtain their fully optimal solution Cogley and Sargent (2008) need to assume a finite planning horizon as well as a two-state Markov process for income, and even then, computation of the optimal decision rule requires a great deal of technical expertise.

The approach taken by Adam and Marcet (2011), like Cogley and Sargent (2008), requires that agents solve a dynamic programming problem given their beliefs. These beliefs take the form of a fully specified distribution over all potential future paths of those variables taken as external to the agents. This is somewhat more general than Cogley and Sargent (2008) in that the distribution may or may not involve parameters that need to be estimated. Adam and Marcet (2011) analyze a basic asset pricing model with heterogeneous agents, incomplete markets, linear utility and limit constraints on stock holding. Within this model, they define an “internally rational” expectations equilibrium (IREE) as characterized by a sequence of pricing functions mapping the fundamental shocks to prices, such that markets clear, given agents’ beliefs and corresponding optimal behavior.

In the Adam and Marcet (2011) approach, agents may be viewed quite naturally as Bayesians, i.e., they may have forecasting models in mind with distributions over the models’ parameters. In this sense agents are adaptive learners in a manner consistent with forming forecasts optimally against the implied conditional distributions obtained from a “well-defined system of subjective probability beliefs.” An REE is an IREE in which agents’ “internal” beliefs are consistent with the external “truth,” that is, with the objective equilibrium distribution of prices. Since they require that, in equilibrium, the pricing function is a map from shocks to prices, it follows that
agents must hold the belief that prices are functions only of the shocks – in this way, REE beliefs reflect a singularity: the joint distribution of prices and shocks is degenerate, placing weight only on the graph of the price function. Their particular set-up has one other notable feature, that the optimal decisions of each agent require only one-step ahead forecasts of prices and dividend. This would not generally hold for risk averse agents, as can be seen from the set-up of Cogley and Sargent (2008), in which a great deal of sophistication is required to solve for the optimal plans.

Using the “anticipated utility” framework, Preston (2005) develops an infinite-horizon (or long-horizon) approach, in which agents use past data to estimate a forecasting model; then, treating these estimated parameters as fixed, agents make time \( t \) decisions that are fully optimal. This decision-making is optimal in the sense that it incorporates the (perceived) lifetime budget constraint (LBC) and the transversality condition (TVC). However, in this approach agents ignore the knowledge that their estimated forecasting model will change over time. Applications of the approach include, for example, Eusepi and Preston (2010) and Evans, Honkapohja, and Mitra (2009). Long-horizon forecasts were also emphasized in Bullard and Duffy (1998).

A commonly used approach known as Euler equation learning, developed e.g. in Evans and Honkapohja (2006), takes the Euler equation of a representative agent as the behavioral primitive and assumes that agents make decisions based on the boundedly rational forecasts required by the Euler equation. As in the other approaches, agents use estimated forecast models, which they update over time, to form their expectations. In contrast to infinite-horizon learning, agents are behaving in a simple fashion, forecasting only one period in advance. Thus they focus on decisions on this margin and ignore their LBC and TVC. Despite these omissions, when Euler equation learning is stable the LBC and TVC will typically be satisfied. Euler-equation learning is usually done in a linear framework. An application that retains the nonlinear features is Howitt and Özak (2009).

Euler equation learning can be viewed as an agent-level justification for “reduced-form learning,” which is widely used, especially in applied work. Under the reduced-form implementation, one starts with the system of expectational difference equations obtained by linearizing and reducing the equilibrium equations implied by RE, and then replaces RE with subjective one-step ahead forecasts based on a suitable linear

---

3 See also Honkapohja, Mitra, and Evans (2013)

4 A finite-horizon extension of Euler-equation learning is developed in Branch, Evans, and McGough (2013).

5 Howitt and Özak (2009) study boundedly optimal decision making in a non-linear consumption/savings model. Within a finite-state model, agents are assumed to use decision rules that are linear in wealth and updated so as to minimize the squared ex-post Euler equation error, i.e. the squared difference between marginal utility yesterday and discounted marginal utility today, accounting for growth. They find numerically that agents quickly learn to use rules that result in small welfare losses relative to the optimal decision rule.

6 An early example is Bullard and Mitra (2002)
forecasting model updated over time using adaptive learning. This approach leads to a particularly simple stability analysis,\textsuperscript{7} but often fails to make clear the explicit connection to agent-level decision making.

The above procedures all involve forecasting, and thus require an estimate of the transition dynamics of the economy. This estimation step can be avoided using an approach called Q-learning, developed originally by Watkins (1989) and Watkins and Dayan (1992). Under Q-learning, which is most often used in finite-state environments, an agent estimates the “quality values” associated with each state/action pair. One advantage of Q-learning is that it eliminates the need to form forecasts by updating quality measures ex-post. To pursue the details some notation will be helpful. Let \( x \in X \) represent a state and \( a \in A \) represent an action. The usual Bellman system has the form 
\[
V(x) = \max_{a \in A} \left( r(x, a) + \beta \sum_{y} P_{xy}(a) V(y) \right),
\]
where \( r \) captures the instantaneous return and \( P_{xy}(a) \) is the probability of moving from state \( x \) to state \( y \) given action \( a \). The quality function \( Q : X \times A \to \mathbb{R} \) is defined as
\[
Q(x, a) = r(x, a) + \beta \sum_{y} P_{xy}(a) V(y).
\]
Under Q-learning, given \( Q_{t-1}(x, a) \), the estimate of the quality function at (the beginning of) time \( t \), and given the state \( x \) at \( t \), the agent chooses the action \( a \) with the highest quality, i.e. \( a = \max_{a' \in A} Q_{t-1}(x, a') \). At the beginning of time \( t + 1 \), the estimate of \( Q \) is updated recursively as follows:
\[
Q_t(x, a) = Q_{t-1}(x, a) + \frac{1}{t} \left( a = \max_{a' \in A} Q_{t-1}(x, a') \right) \left( r(x, a) + \beta \max_{b \in A} Q_{t-1}(y, b) \right),
\]
where \( I \) is the indicator function and \( y \) is the state that is realized in \( t + 1 \). Notice that \( Q_t \) does not require knowledge of the state’s transition function. Provided the state and action spaces are finite, Watkins and Dayan (1992) show \( Q_t \to Q \) almost surely under a key assumption, which requires in particular each state/action pair is visited infinitely many times. We note that this assumption is not easily generalized to the continuous state and action spaces that are standard in the macroeconomic literature.

A related approach to boundedly rational decision making uses classifier systems. An early well-known economic application is Marimon, McGrattan, and Sargent (1990). They introduce classifier system learning into the model of money and matching due to Kiyotaki and Wright (1989). They consider two types of classifier systems. In the first, there is a complete enumeration of all possible decision rules. This is possible in the Kiyotaki-Wright set-up because of the simplicity of that model. The second type of classifier system instead uses rules that do not necessarily distinguish each state, and which uses genetic algorithms to periodically prune rules and generate new ones. Using simulations Marimon et al. show that learning converges to a

\textsuperscript{7}See Chapter 10 of Evans and Honkapohja (2001).
stationary Nash equilibrium in the Kiyotaki-Wright model, and that, when there are 
multiple equilibria, learning selects the fundamental low-cost solution.

Lettau and Uhlig (1999) incorporate rules of thumb into dynamic programming 
using classifier systems. In their “general dynamic decision problem” they consider 
agents maximizing expected discounted utility, where agents make decisions using 
rules of thumb (a mapping from a subset of states into the action space, giving a 
specified action for specified states within this subset). Each rule of thumb has an 
associated strength. Learning takes place via updating of strengths. At time \( t \) the 
classifier with highest strength among all applicable classifiers is selected and the 
corresponding action is undertaken. After the return is realized and the state in \( t + 1 \) 
is (randomly) generated, the strength of the classifier used in \( t \) is updated (using a 
gain sequence) by the return plus \( \beta \) times the strength of the strongest applicable 
classifier in \( t + 1 \). Lettau and Uhlig give a consumption decision example, with 
two rules of thumb, the optimal decision rule based on dynamic programming and 
another non-optimal rule of thumb, applicable only in high-income states, in which 
agents consume all their income. They showed that convergence to this suboptimal 
rule of thumb is possible.\(^8\)

Our SP-learning framework shares various characteristics of the alternative imple-
mentations of agent-level learning discussed above. Like Q-learning and the related 
approaches based on classifier systems, SP-learning builds off of the intuition of the 
Bellman equation. (In fact, what we will call value-function learning explicitly es-

tablishes the connection.) As in infinite-horizon learning, we employ the anticipated 
utility approach rather than the more sophisticated Bayesian perspective. Like Euler-
equation learning, it is sufficient for agents to look only one step ahead. While each 
of the alternative approaches has advantages, we find SP-learning persuasive in many 
applications due to its simplicity, generality and economic intuition.

2.2 Shadow-price learning

Returning to the current paper, our objective is to develop a general approach for 
boundedly rational decision-making in a dynamic stochastic environment. While par-
ticular examples would include the optimal consumption-savings problems summa-
rized above, the technique is generally applicable and can be embedded in standard 
general equilibrium macro models. To illustrate our technique, consider a standard 

\(^8\)Lettau and Uhlig discuss the relationship of their decision rule to Q-learning in their footnote 
11, p. 165: they state that (i) Q-learning also introduces action mechanisms that ensure enough 
exploration so that all \((x, a)\) combinations are triggered infinitely often, and (ii) in Q-learning the 
value \( Q(x, a) \) is assigned and updated for every state-action pair \((x, a)\). This corresponds to classifiers 
that are only applicable in a single state. In general, classifiers are allowed to cover more general 
sets of state-action pairs.
dynamic programming problem

\[ V(x_0) = \max E_0 \sum_{t \geq 0} \beta^t r(x_t, u_t) \]

subject to \( x_{t+1} = g(x_t, u_t, \xi_{t+1}) \)

and \( x_0 \) given. Here \( u_t \in \Gamma(x_t) \subseteq R^m \) is the vector of controls (with \( \Gamma(x_t) \) compact), \( x_t \in R^n \) is the vector of (endogenous and exogenous) states variables, and \( \xi_{t+1} \) is white noise. Our approach is based on the standard first-order conditions derived from the Lagrangian\(^9\)

\[ L = E_0 \sum_{t \geq 0} \beta^t (r(x_t, u_t) + \lambda_t (g(x_{t-1}, u_{t-1}, \xi_t) - x_t)) , \]

namely

\[ \lambda_t = r_x(x_t, u_t)' + \beta E_t g_x(x_t, u_t, \xi_{t+1})' \lambda_{t+1} \]
\[ 0 = r_u(x_t, u_t)' + \beta E_t g_u(x_t, u_t, \xi_{t+1})' \lambda_{t+1}. \]

Under the SP-learning approach we replace \( \lambda_t \) with \( \lambda_t^* \), representing the perceived shadow price of the state, and we treat equations (3)-(4) as the basis of a behavioral decision rule.

To implement SP-learning (3)-(4) need to be supplemented with forecasting equations for the required expectations. In line with the adaptive learning literature, assume that the transition equation (2) is unknown, and must be estimated, and that agents do so by approximating the transition equation using a linear specification of the form\(^10\)

\[ x_{t+1} = Ax_t + Bu_t + C\xi_{t+1}, \]

and thus the agents approximate \( g_x(x_t, u_t, \xi_{t+1}) \) by \( A \) and \( g_u(x_t, u_t, \xi_{t+1}) \) by \( B \). The coefficient matrices \( A, B \) are estimated and updated over time using recursive least squares (RLS). We also assume that agents believe the perceived shadow price \( \lambda_t^* \) is (or can be approximated by) a linear function of state, up to white noise, i.e.

\[ \lambda_t^* = Hx_t + \mu_t, \]

where the matrix \( H \) also is estimated. Finally, we assume that agents know their preference function \( r(x_t, u_t) \). Then, given the state \( x_t \) and estimates for \( A, B, H \) the decision procedure is obtained by solving the system

\[ r_u(x_t, u_t)' = -\beta B' \hat{E}_t \lambda_{t+1}^* \]
\[ \hat{E}_t \lambda_{t+1}^* = H (Ax_t + Bu_t) \]

\(^9\)For \( t = 0 \) the last term in the sum is replaced by \( \lambda_0^*(\bar{x}_0 - x_0) \), where \( \bar{x}_0 \) is the initial state vector.
\(^10\)Here we have expanded the state vector \( x_t \) to include a constant.
for $u_t$ and the forecasted shadow price, $\hat{E}_t\lambda^*_{t+1}$. These values can then be used with (3) to obtain an updated estimate of the current shadow price

$$\lambda^*_t = r_x(x_t, u_t)' + \beta A'\hat{E}_t\lambda^*_{t+1}. \tag{6}$$

Finally, the data $(x_t, u_t, \lambda^*_t)$ can be used to recursively update the parameter estimates $(A, B, H)$ over time. Taken together this procedure defines a natural implementations of the SP-learning approach.

As we will see, under more specific assumptions, this implementation of boundedly optimal decision making leads to asymptotically optimal decisions. In this sense shadow-price learning is reasonable from an agent perspective. Our approach has a number of strengths. Particularly attractive, we think, is the pivotal role played by shadow prices. In economics prices are central because agents use them to assess trade-offs. Here the perceived shadow price of next period’s state vector, together with the estimated transition dynamics, measures the intertemporal trade-offs and thereby determines the agent’s choice of control vector today. The other feature that we find compelling is the simplicity of the required behavior: agents make decisions as if they face a two-period problem. In this way we eliminate the discontinuity between the sophistication of agents as forecasters and agents as decision-makers. In addition, SP-learning incorporates the RLS updating of parameters that is the hallmark of the adaptive learning approach. Finally, this version of bounded optimality is applicable to the general stochastic regulator problem, and can be embedded in standard DSGE models.

While we view SP-learning as a very natural implementation of bounded optimality, there are some closely related variations that also yield asymptotic optimality. In Section 6 of his seminal paper on asset pricing, discussing stability analysis, Lucas (1978) briefly outlines how agents might update over time their subjective value function. In Section 3.4 we show how to specify a real-time procedure for updating an agent’s value function. Our Theorem 6 implies that this procedure converges asymptotically to the true value function of an optimizing agent. From Section 3.4 it can seen that another variation, Euler-equation learning, is in some cases equivalent to SP-learning. Indeed, Theorem 7 establishes the first formal general convergence result for Euler-equation learning by establishing its connections to SP-learning.

Having found that SP-learning is reasonable from an agent’s perspective, in that he can expect to eventually behave optimally, we embed shadow price learning into a simple economy consistent with our quadratic regulator environment. We consider a Robinson Crusoe economy, with quadratic preferences and linear technology: see Hansen and Sargent (2014) for many examples of these types of economies, including one of the examples we give. By including production lags we provide a simple example of a multivariate model in which SP-learning and Euler-equation learning differ. We use the Crusoe economy to walk carefully through the boundedly optimal behavior displayed by our agent, thus providing examples of, and intuition for the
behavioral assumptions made in Section 3.1.

While our formal results are proved for the Linear-Quadratic framework, as we have stressed, the techniques and intuition can be applied in a general setting. To illustrate this point we conclude with an application to the Ramsey model, in which we impose that the representative agent act as an SP learner within a general equilibrium environment. We show that asymptotically the household’s perceived shadow price approximates the corresponding social planner’s Lagrange multiplier.

3 Learning to optimize

We begin be specifying the programming problem of interest. We focus on the behavior of a decision maker with a quadratic objective function and who faces a linear transition equation; the linear-quadratic (LQ) set-up allows us to exploit certainty equivalence and to conduct parametric analysis. The specification of our LQ problem, which is standard, is taken from Hansen and Sargent (2014); see also Stokey and Lucas Jr. (1989), and Bertsekas (1987).

3.1 Linear quadratic dynamic programing

The “sequence problem” is to determine a sequence of controls $u_t$ that solves

$$
\begin{align*}
\max_{u_t} \quad & -E_0 \sum \beta^t (x'_t Rx_t + u'_t Qu_t + 2x'_t Wu_t) \\
\text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t + C \varepsilon_{t+1},
\end{align*}
$$

with $x_0$ taken as given. As with the general dynamic programming problem (1) - (2), we assume $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, with the matrices conformable. We further assume that $\varepsilon_t \in \mathbb{R}^k$ is a zero-mean i.i.d. process with $E \varepsilon_t \varepsilon'_t = \sigma^2 \delta I$ and compact support. The assumptions on $\varepsilon_t$ are convenient but can be relaxed considerably: for example, Theorem 1 only requires that $\varepsilon_t$ be a martingale difference sequence with (finite) time-invariant second moments; and, Theorem 3 holds if the assumption of compact support is replaced by the existence of finite absolute moments.

Under well-known conditions that will be discussed in detail below, this problem has a unique solution, and the sequence of controls are determined by $u_t = F^* x_t$ for an appropriate feedback matrix $F^*$. This matrix is obtained by analyzing the associated Bellman functional equation:

$$
V(x) = \max_u - (x' Rx + u' Qu + 2x' Wu) + \beta E (V(Ax + Bu + C \varepsilon)|x, u).
$$

11 The LQ set-up can also be used to approximate more general nonlinear environments.
By guessing that $V$ has a quadratic functional form, it is not difficult to show that

\[ V(x) = -x'P^*x - \text{tr} \left( \sigma_x^2 P^* C' \right), \tag{8} \]

\[ F^* = - (Q + \beta B'P^*B)^{-1} \left( \beta B'P^*A + W' \right), \tag{9} \]

where \( \text{tr} \) denotes the trace of a matrix and where, again under appropriate assumptions, \( P^* \) is the unique symmetric positive semi-definite matrix satisfying the Riccati equation

\[ P^* = T(P^*), \tag{10} \]

\[ T(P) = R + \beta A'PA - (\beta A'PB + W)(Q + \beta B'PB)^{-1}(\beta B'PA + W'). \tag{11} \]

Solving the Riccati equation is not possible analytically; however, a variety of numerical methods are available.

Conditions sufficient to guarantee the problem (7) has a unique solution and that the Riccati equation has a unique positive semi-definite solution are most easily stated by transforming the problem to eliminate the state-control interaction in the objective and discounting: see Hansen and Sargent (2014), Chapter 3 for the many details. Set $\hat{A} = \beta^{\frac{1}{2}}(A - BQ^{-1}W'$), $\hat{B} = \beta^{\frac{1}{2}}B$, and $\hat{R} = R - WQ^{-1}W'$. Letting $\hat{x}_t = \beta^{\frac{1}{2}}x_t$, the problem becomes

\[ \max \quad -E_0 \sum_t \left( \hat{x}'_t R \hat{x}_t + u'_t Q u_t \right) \tag{12} \]

\[ \text{s.t.} \quad \hat{x}_{t+1} = \hat{A} \hat{x}_t + \hat{B} u_t + \beta^{\frac{t+1}{2}}C \varepsilon_{t+1}. \]

To place restrictions on the matrices identifying the problem (12), a few definitions are needed. Below (in LQ.1) we assume that $R$ is symmetric positive semidefinite. Using the rank decomposition, $R$ can be factored as $R = \hat{D} \hat{D}'$, where rank($R$) = $r$ and $\hat{D}$ is $n \times r$. \(^\text{12}\)With this notation, we say that:

- A matrix is stable if its eigenvalues have modulus less than one.
- The matrix pair $(\hat{A}, \hat{B})$ is stabilizable if there exists a matrix $K$ such that $\hat{A} + \hat{B}K$ is stable.
- The matrix pair $(\hat{A}, \hat{D})$ is detectable provided that whenever $y$ is a (nonzero) eigenvector of $\hat{A}$ associated with the eigenvalue $\lambda$ and $\hat{D}'y = 0$ it follows that $|\lambda| < 1$. Intuitively, $\hat{D}'$ acts as a factor of the objective function’s quadratic form $\hat{R}$: if $\hat{D}'y = 0$ then $y$ is not detected by the objective function; in this case, the associated eigenvalue must be contracting.

\(^{12}\)Any positive semi-definite matrix $X$ may be factored as $X = UAU'$, where $U$ is a unitary matrix. The rank decomposition $X = \hat{D} \hat{D}'$ obtains by writing $\Lambda = \Lambda_1 \oplus 0$, with $\Lambda_1$ invertible, and letting $\hat{D} = (U'_{11}, U'_{21})' \sqrt{\Lambda_1}$. 12
The matrix pair \( (\hat{A}, \hat{R}) \) is observable if whenever \( y \) is an eigenvector of \( \hat{A} \) it follows that \( \hat{R}y \neq 0 \). Intuitively, every eigenvector of \( \hat{A} \) must impact the objective, i.e. be “observed” by \( \hat{R} \). We note that if \( (\hat{A}, \hat{R}) \) is observable then \( (\hat{A}, \hat{D}) \) is detectable.

LQ.1: The matrix \( \hat{R} \) is symmetric positive semi-definite and the matrix \( Q \) is symmetric positive definite.

LQ.2: The system \( (\hat{A}, \hat{B}) \) is stabilizable.

LQ.3: The system \( (\hat{A}, \hat{R}) \) is observable.

This list represents a standard set of assumptions sufficient to guarantee a well-behaved problem. LQ.1 imparts the appropriate concavity assumptions on the objective and LQ.2 says that it is possible to find a set of controls driving the state to zero in the deterministic problem. The need for LQ.3 is slightly more subtle: if \( (\hat{A}, \hat{R}) \) is observable then \( (\hat{A}, \hat{D}) \) is detectable and the control path must be chosen to counter dynamics in the explosive eigenspaces of \( \hat{A} \). To illustrate, suppose \( z \) is an eigenvector of \( A \) with associated eigenvalue \( \lambda \), suppose that \( |\lambda| > 1 \), and finally assume that \( x_0 = z \). If the control path is not chosen to mitigate the explosive dynamics in the eigenspace associated to \( \lambda \) then the state vector will diverge in norm. Furthermore, because \( (\hat{A}, \hat{D}) \) is detectable, we know that \( \hat{D}'z \neq 0 \). Taken together, these observations imply that an explosive state is suboptimal:

\[
-x_t' \hat{R} x_t = -\lambda^{2t} z' \hat{D} \hat{D}' z = -\left( |\lambda|^{t} |\hat{D}' z| \right)^2 \to -\infty.
\]

Hansen and Sargent (see Appendix A of Ch. 3 in Hansen and Sargent (2014)) put it more concisely (and eloquently): If \( (\hat{A}, \hat{B}) \) is stabilizable then it is feasible to stabilize the state vector; if \( (\hat{A}, \hat{D}) \) is detectable then it is desirable to stabilize the state vector. While the detectability of \( (\hat{A}, \hat{D}) \) is sufficient to guarantee that the Riccati equation (10) has a unique positive semi-definite solution, the link between this solution and the control problem requires a slightly stronger condition. As is standard in the literature, we adopt LQ.3 (observability of \( (\hat{A}, \hat{R}) \)) as a benchmark assumption. However, in the Appendix we provide a weaker alternative assumption LQ.3’, based on observability of the subsystem corresponding to the endogenous states, which is convenient for applications.

**Theorem 1** Under assumptions LQ.1 – LQ.3, the Riccati equation (10) has a unique positive semi-definite solution, \( P^* \), and iteration of the Riccati equation yields convergence to \( P^* \) if initialized at any positive semi-definite matrix \( P_0 \). Also, there is a unique sequence of controls solving (7), and they are given by \( u_t = F^* x_t \), where \( F^* \) is determined by (9).
This theorem is well known and follows from results in Section 2 and the Stability Theorem in Section 4, Chapter 3, of Hansen and Sargent (2014). See also Theorem 16.6.4, on page 368 of Lancaster and Rodman (1995).

We now turn to our workhorse result, Lemma 2 below, on which our stability results are based. We begin by securing the notation needed to compute derivatives when we have matrix-valued functions and matrix-valued differential equations. If \( f : \mathbb{R}^p \to \mathbb{R}^q \) then \( D(f) \) is the matrix of first partials, and for \( x \in \mathbb{R}^p \), the notation \( D(f)(x) \) emphasizes that the partials are evaluated at the vector \( x \). Notice that \( D \) is an operator that acts on vector-valued functions – we do not apply \( D \) to matrix-valued functions. The analysis of matrix-valued differential equations is conducted by working through the vec operator.13 If \( f : \mathbb{R}^{p \times q} \to \mathbb{R}^{p \times q} \) then we define \( f_v : \mathbb{R}^p \to \mathbb{R}^{p^2} \) by \( f_v = vec \circ f \circ vec^{-1} \), where the dimensions of the domain and range of the vec operators employed are understood to be determined by \( f \). Thus suppose \( f : \mathbb{R}^{p \times p} \to \mathbb{R}^{p \times p} \), assume \( f(x^*) = 0 \), and consider the matrix-valued differential equation \( \dot{x} = f(x) \), where \( \dot{x} \) denotes the derivative with respect to time. Let \( y = vec(x) \), and note that \( \dot{y} = vec(\dot{x}) \). Then

\[
\dot{x} = f(x) \implies vec(\dot{x}) = vec(f(x)) \implies \dot{y} = (vec \circ f \circ vec^{-1})(vec(x)) \implies \dot{y} = f_v(y).
\]

Hence if \( y^* = vec(x^*) \) then Lyapunov stability of \( x^* \) may be assessed by determining the eigenvalues of \( D(f_v)(y^*) \).

With this notation in hand, we are ready to present and prove our workhorse lemma.

**Lemma 2** Assume LQ.1, LQ.2, that \((\hat{A}, \hat{D})\) is detectable, and that \(P^*\) is the unique positive semi-definite solution to the Riccati equation. Then the eigenvalues of the Jacobian matrix \(D(T_v(vec(P^*)))\) are strictly inside the unit circle.

We note that the observability of \((\hat{A}, \hat{R})\) implies the detectability of \((\hat{A}, \hat{D})\), so that LQ.1 – LQ.3 imply the premises of the Lemma are met. Furthermore, we note that LQ.1, LQ.2, and the assumption that \((\hat{A}, \hat{D})\) is detectable implies the existence of a unique positive semi-definite solution to the Riccati equation: see Lancaster and Rodman (1995), Theorem 13.5.2. Lemma 2 lies at the heart of our stability results, not only for SP-learning, but also for variations of our approach.

---

13 The vec operator is the standard isomorphism coupling \( \mathbb{R}^{n \times m} \) with \( \mathbb{R}^{nm} \). Intuitively, the vectorization of a matrix \( Z \) is obtained by simply stacking its columns. More formally, let \( Z \in \mathbb{R}^{n \times m} \). For each \( 1 \leq k \leq nm \), use the division algorithm to uniquely write \( k = jn + i \), for \( 0 \leq j \leq m \) and \( 0 \leq i < n \). Then

\[
vec(Z)_k = \begin{cases} 
Z_{1j} & \text{if } j = 0 \\
Z_{nj} & \text{if } i = 0 \\
Z_{i,j+1} & \text{else}
\end{cases}
\]
3.2 Bounded optimality: shadow-price learning

For an agent to solve the programming problem (7) as described above, he must understand the quadratic nature of his value function as captured by the matrix $P^*$; he must know the relationship of this matrix to the Riccati equation, he must be aware that iteration on the Riccati equation provides convergence to $P^*$, and finally, he must know how to deduce the optimal control path given $P^*$. Furthermore, this behavior is predicated upon the assumption that he knows the conditional means of the state variables, that is, he knows $A$ and $B$.

We modify the primitives identifying agent behavior, first by imposing bounded rationality and then by assuming bounded optimality. Our agent is not assumed to know the state variables’ conditional means: he must estimate $A$ and $B$. Our agent is also not assumed to know how to solve his programming problem: he does not know Theorem 1. Instead, he uses a simple forecasting model to estimate the value of a unit of state tomorrow, and then he uses this forecast, together with his estimate of the transition equation, to determine his control today. Based on his control choice and his forecast of the value of a unit of state tomorrow, he revises the value of a unit of state today. This provides him new data to update his state-value forecasting model.

We develop our analysis of the agent’s boundedly rational behavior in two stages. In the first stage, which we call “stylized learning,” we avoid the technicalities introduced by the stochastic nature of data realization and forecast-model estimation; instead, we simply assume that the agent’s beliefs evolve according to a system of differential equations – the E-stability equations – which, we will argue, have a natural, intuitive appeal. In the second stage, we will then formally connect the E-stability equations to the asymptotic dynamics of the agent’s beliefs under the assumption that he is recursively estimating his forecasting models and behaving accordingly.

3.2.1 Stylized shadow-price learning

To facilitate intuition for our learning mechanism, we reconsider the above problem using a Lagrange multiplier formulation. The Lagrangian is given by

$$\mathcal{L} = E_0 \sum_{t \geq 0} \beta^t \left( -x'_t R x_t - u'_t Q u_t - 2x'_t W u_t + \lambda'_t (Ax_{t-1} + Bu_{t-1} + C\varepsilon_t - x_t) \right),$$

where again for $t = 0$ the last term in the sum is replaced by $\lambda'_0(\bar{x}_0 - x_0)$. As usual, $\lambda_t$ may be interpreted as the shadow price of the state vector $x_t$ along the optimal path. The first-order conditions provide

$$\mathcal{L}_{x_t} = 0 \Rightarrow \lambda'_t = -2x'_t R - 2u'_t W' + \beta E_t \lambda'_{t+1} A$$
$$\mathcal{L}_{u_t} = 0 \Rightarrow 0 = -2u'_t Q - 2x'_t W + \beta E_t \lambda'_{t+1} B.$$
Transposing and combining with the transition equation yields the following dynamic system:

\[
\begin{align*}
\lambda_t &= -2Rx_t - 2Wu_t + \beta A' E_t \lambda_{t+1} \\
0 &= -2W'x_t - 2Qu_t + \beta B' E_t \lambda_{t+1} \\
x_{t+1} &= Ax_t + Bu_t + C\varepsilon_{t+1},
\end{align*}
\]

This system, together with transversality, identifies the unique solution to (7). It also provides intuitive behavioral restrictions on which we base our notion of bounded optimality.

We now marry the assumption from the learning literature that agents make boundedly rational forecasts with a list of behavioral assumptions characterizing the decisions agents make given these forecasts; and, we do so in a manner that we feel imparts a level of sophistication consistent with bounded rationality. Much of the learning literature centers on equilibrium dynamics implied by one-step-ahead boundedly rational forecasts; we adopt and expand on this notion by developing assumptions consistent with the following intuition: agents make one-step-ahead forecasts and agents know how to solve a two-period optimization problem based on their forecasts. Formalizing this intuition, we make the following assumptions:

1. Agents know their individual instantaneous return function, that is, they know $Q$, $R$, and $W$;
2. Agents know the form of the transition law and estimate the coefficient matrices;
3. Conditional on their perceived value of an additional unit of $x$ tomorrow, agents know how to choose their control today;
4. Conditional on their perceived value of an additional unit of $x$ tomorrow, agents know how to compute the value of an additional unit of $x$ today.

Assumption one seems quite natural: if the agent is to make informed decisions about a certain vector of quantities $u$, he should at least be able to understand the direct impact of these decisions. Assumption two is standard in the learning literature: our agent needs to forecast the state vector, but is uncertain about its evolution; therefore, he specifies and estimates a forecasting model, which we take as having the same functional form as the linear transition equation, and forms forecasts accordingly. Denote by $\hat{A}$ and $\hat{B}$ the agent’s perceptions of $A$ and $B$ respectively. As will be discussed below, under stylized learning, these perceptions are assumed to evolve over time according to a differential equation, whereas under real-time learning, the agent’s perceptions are taken as estimates which he updates as new data become available.
Assumptions three and four require more explanation. Let $\lambda_t^*$ be the agent’s perceived shadow price of $x_t$ along the realized path of $x$ and $u$. One should not think of $\lambda^*$ as identical to $\lambda$; indeed $\lambda$ is the vector of shadow prices of $x$ along the optimal path of $x$ and $u$ and the agent is not (necessarily) interested in this value. Let $\hat{E}_t \lambda_{t+1}^*$ be the agent’s time $t$ forecast of the time $t+1$ value of an additional increment of the state $x$. Assumption three says that given $\hat{E}_t \lambda_{t+1}^*$, the agent knows how to choose $u_t$, that is, he knows how to solve the associated two-period problem. And how is this choice made? The agent simply contemplates an incremental decrease $\delta u$ and equates marginal loss with marginal benefit. If $\rho$ is the “rate function” $\rho(x, u) = - (x'R x + u'Q u + 2x'W u)$ then the marginal loss is $r_{u_t} du_t$. To compute the marginal gain, he must estimate the effect of $\delta u$ on the whole state vector tomorrow. This effect is determined by $\hat{B}_t du_t$, where $\hat{B}_t$ is the $t^{th}$-column of the beliefs matrix $\hat{B}$. To weigh this effect against the loss obtained in time $t$, he must then compute its inner product with the expected price vector, and discount. Thus

$$r_{u_t} du_t = \beta \hat{E}_t (\lambda_{t+1}^*)' \hat{B}_t du_t.$$  

Stacking, and imposing our linear-quadratic set-up, gives the bounded rationality equivalent to (14):

$$0 = -2W'x_t - 2Q u_t + \beta \hat{B}' \hat{E}_t \lambda_{t+1}^*.$$  

Equation (16) operationalizes assumption three.

To update their shadow-price forecasting model, the agent needs to determine the perceived shadow price $\lambda_t^*$. Assumption four says that given $\hat{E}_t \lambda_{t+1}^*$, the agent knows how to compute $\lambda_t^*$. And how is this price computed? The agent simply contemplates an incremental increase in $x_{it}$ and evaluates the benefit. An additional unit of $x_{it}$ affects the contemporaneous return and the conditional distribution of tomorrow’s state; and the shadow price $\lambda_{it}^*$ must encode both of these effects. Specifically, if $r$ is the rate function then the benefit of $dx_{it}$ is given by

$$r_{x_t} dx_{it} + \beta \hat{E}_t (\lambda_{t+1}^*)' \hat{A}_t dx_{it} = \lambda_{it}^* dx_{it}$$

where the equality provides our definition of $\lambda_{it}^*$. Stacking, and imposing our linear quadratic set-up yields the bounded rationality equivalent to (13):

$$\lambda_t^* = -2Rx_t - 2W_{it} u_t + \beta \hat{A}' \hat{E}_t \lambda_{t+1}^*.$$  

Equation (17) operationalizes assumption four.

Assumption three, as captured by (16), lies at the heart of bounded optimality: it provides that the agent makes one-step-ahead forecasts of shadow prices and makes decisions today based on those forecasted prices, just as he would if solving a two-period problem. Assumption four, as captured by (17), provides the mechanism by which the agent computes his revised evaluation of a unit of state at time $t$: the agent
uses the forecast of prices at time $t + 1$ and his control decision at time $t$ to reassess the value of time $t$ state; in this way, our boundedly optimal agent keeps track of his forecasting performance. Below, the agent uses $\lambda_t^*$ to update his shadow-price forecasting model. We call boundedly optimal behavior, as captured by assumptions one through four, **shadow price learning**.

We now specify the shadow-price forecasting model, that is, the way our agent forms $\hat{E}_t \lambda_{t+1}^*$. Along the optimal path it is not difficult to show that $\lambda_t = -2P^* x_t$, and so it is natural to impose a forecasting model of this functional form. Therefore, we assume that at time $t$ the agent believes that

$$\lambda_t^* = H x_t + \mu_t$$

for some $n \times n$ matrix $H$ (which we assume is near $-2P^*$) and some error term $\mu_t$. Equation (18) has the feel of what is known in the learning literature as a perceived law of motion (PLM): the agent perceives that his shadow price exhibits a linear dependence on the state as captured by the matrix $H$. When engaged in real-time decision making, as considered in Section 3.2.2, our agent will revisit his belief $H$ as new data become available. Under our stylized learning mechanism, $H$ is taken to evolve according to a differential equation as discussed below.

We can now be precise about the agent’s behavior. Given beliefs $\tilde{A}$, $\tilde{B}$ and $H$, expectations are formed using (18):

$$\hat{E}_t \lambda_{t+1}^* = H (\tilde{A} x_t + \tilde{B} u_t).$$

Equations (16) and (19) jointly determine the agent’s time $t$ forecast $\hat{E}_t \lambda_{t+1}^*$ and time $t$ control decision $u_t$. Finally, (17) is used to determine the agent’s perceived shadow price of the state $\lambda_t^*$.

It follows that the evolution of $u_t$ and $\lambda_t^*$ satisfy

$$u_t = (2Q - \beta \tilde{B}' H \tilde{B}^{-1} (\beta \tilde{B}' H \tilde{A} - 2W') x_t$$

$$\equiv F(H, \tilde{A}, \tilde{B}) x_t,$$

and

$$\lambda_t^* = \left(-2R - 2W F(H, \tilde{A}, \tilde{B}) + \beta \tilde{A}' H (\tilde{A} + \tilde{B} F(H, \tilde{A}, \tilde{B}))\right) x_t$$

$$\equiv T^{SP}(H, \tilde{A}, \tilde{B}) x_t,$$

where the second lines define notation.\(^{14}\) Note that it is not necessary to assume, nor do we assume, that our agent computes the map $T^{SP}$.

We now turn to stylized learning, which dictates how our agent’s beliefs, as summarized by the collection $(H, \tilde{A}, \tilde{B})$, evolve over time. In order to draw comparisons

\(^{14}\)Since $Q$ is positive definite and $P^*$ is positive semi-definite, it follows that $Q + \beta \tilde{B}' P^* \tilde{B}$ is invertible; thus $2Q - \beta \tilde{B}' H \tilde{B}$ is invertible for $H$ near $-2P^*$.  

18
and promote intuition for our learning model, it is useful first to succinctly summarize the corresponding notion from the macroeconomics adaptive learning literature. There, boundedly rational forecasters are typically assumed to form expectations using a forecasting model, or PLM, that represents the believed dependence of relevant variables (i.e., variables that require forecasting) on regressors; the actions these agents take then generate an implied dependence, or actual law of motion (ALM), and the map taking perceptions, say as summarized by a vector $\Theta$, to the implied dynamics, is denoted with a $T$. This T-map captures the model’s “expectational feedback” in that it measures how agents’ perceptions of the relationship between the relevant variables and the regressors feeds back to the realized relationship.

A fixed point of the T-map, say $\Theta^*$, corresponds to a rational expectations equilibrium (REE): agents’ perceptions coincide with the true data-generating process, so that expectations are being formed optimally. Conversely, the discrepancy between perceptions and reality, as measured by $T(\Theta) - \Theta$, captures the direction and magnitude of the misspecification in agents’ beliefs. Under a stylized learning mechanism, agents are assumed to modify their beliefs in response to this discrepancy according to the expectational stability (E-stability) equation $d\Theta / d\tau = T(\Theta) - \Theta$. Notice that $\Theta^*$ represents a fixed point of this differential equation. If this fixed point is Lyapunov stable then the corresponding REE is said to be E-stable, and stability indicates that an economy populated with stylized learners would eventually be in the REE.\(^{15}\)

Returning to our environment in which a single agent makes boundedly rational decisions, we observe that shadow-price learning is quite similar to the model of boundedly rational forecasting just discussed. Equation (18) has already been interpreted as a PLM, and equation (22) acts as an ALM and reflects the model’s feedback: the agent’s beliefs, choices and evaluations result in an actual linear dependence of his perceived shadow price on the state vector as captured by $T^{SP}(H, \hat{A}, \hat{B})$. Notice that the actual dependence of $x_{t+1}$ on $x_t$ and $u_t$ is independent of the agent’s perceptions and actions: there is no feedback along these beliefs’ dimensions. For this reason, we assume for now that $\hat{A} = A$ and $\hat{B} = B$, and, abusing notation, we suppress the corresponding dependency in the T-map: $T^{SP}(H) \equiv T^{SP}(H, A, B)$. We will return to this point when we consider real-time learning in Section 3.2.2.

Letting $H^* = -2P^*$, it follows that $T^{SP}(H^*) = H^*$. With beliefs $H^*$, our agent correctly anticipates the dependence of his perceived shadow price on the state vector; also, since

\[
F(H^*, A, B) = -(Q + \beta B' P^* B)^{-1} (\beta B' P^* A + W'),
\]

it follows from equation (9) that with beliefs $H^*$, our agent makes control choices

\(^{15}\)Besides having an intuitive appeal, stylized learning is closely connected to real-time learning through the E-stability Principle, which states that E-stable REE are (locally) learning via recursive least-squares or related algorithms. Formally establishing that the E-stability Principle holds for a given model requires the theory of stochastic recursive algorithms, as discussed and employed in Section 3.2.2 immediately below.
optimally: a fixed point of the map $T^{SP}$ corresponds to optimal beliefs and associated behaviors. Conversely, the discrepancy between the perceived and realized dependence of $\lambda^*_t$ on $x_t$, as measured by $T^{SP}(H) - H$, captures the direction and magnitude of the misspecification in our agent’s beliefs. Whereas in the literature on adaptive learning this discrepancy arises because the agent does not fully understand the conditional distributions of economy’s aggregate variables, here the discrepancy reflects our agent’s limited sophistication: he does not fully understand his dynamic programming problem, and instead bases his decisions on his best measure of the trade-offs he faces, as reflected by his belief matrix $H$.

In Theorem 3 we embrace the concept of stylized learning presented above and assume our agent updates his beliefs according to the matrix-valued differential equation

$$ dH/d\tau = T^{SP}(H) - H. $$

Note that $H^*$ is a fixed point of this dynamic system. The following theorem together with Theorem 4, which demonstrates stability under real-time learning, constitute the core results of the paper.

**Theorem 3** If LQ.1 – LQ.3 are satisfied then $H^*$ is a Lyapunov stable fixed point of $dH/d\tau = T^{SP}(H) - H$.

The proof of Theorem 3, which is given in the Appendix, simply involves connecting the maps $T$ and $T^{SP}$, and then using Lemma 2 to assess stability. While Theorem 3 provides stylized learning environment, the main result of our paper, captured by Theorem 4, considers real-time learning. In essence Theorem 4 shows that the stability result of Theorem 3 carries over to the real-time learning environment. We remark that it is easily seen from the proof that assumption LQ.3 in Theorem 3 can be weakened to detectability of $(\hat{A}, \hat{D})$.

### 3.2.2 Real-time shadow-price learning

To establish that stability under stylized learning carries over to stability under real-time learning, we now assume that our agent uses available data to estimate his forecasting model, and then uses this estimated model to form forecasts and make decisions, thereby generating new data. The forecasting model may be written

$$ x_{t+1} = A_t x_t + B_t u_t + \text{error} $$

$$ \lambda^*_t = H_t x_t + \text{error}, $$

where the coefficient matrices are time $t$ estimates obtained using recursive least-squares (RLS).\(^{16}\) We assume that to obtain the estimates $A_t$ and $B_t$, our agent

\(^{16}\)An alternative to RLS is “constant gain” learning (CGL), which discounts older data. Under CGL asymptotic results along the lines of the following Theorem provide for weak convergence to a distribution centered on optimal behavior. See Ch. 7 of Evans and Honkapohja (2001) for general results, and for applications and results concerning transition dynamics, see Williams (2014) and Cho, Williams, and Sargent (2002).
regresses $x_t$ on $x_{t-1}, u_{t-1}$, using data $\{x_t, x_{t-1}, u_{t-1}, \ldots, x_0, u_0\}$.\textsuperscript{17} To estimate the shadow-price forecasting model at time $t$, and thus obtain the estimate $H_t$, we assume our agent regresses $\lambda^*_t$ on $x_{t-1}$ using data $\{x_{t-1}, \ldots, x_0, \lambda^*_{t-1}, \ldots, \lambda^*_0\}$.

We may describe the evolution of the estimate of $A_t, B_t$ and $H_t$ over time using RLS. The following dynamic system, written in recursive causal ordering, captures the evolution of agent behavior under bounded optimality:\textsuperscript{18}

\begin{align*}
x_t &= Ax_{t-1} + Bu_{t-1} + C\varepsilon_t \\
R_t &= R_{t-1} + \frac{1}{t}(x_t x'_t - R_{t-1}) \\
H_t &= H_{t-1} + \frac{1}{t} R^{-1}_{t-1} x_{t-1} (\lambda^*_{t-1} - H_{t-1} x_{t-1})' \\
\hat{R}_t &= \hat{R}_{t-1} + \frac{1}{t} \left( \left( \begin{array}{c} x_{t-1} \\ u_{t-1} \end{array} \right) - \left( \begin{array}{c} A_{t-1} \\ B_{t-1} \end{array} \right) \left( \begin{array}{c} x_{t-1} \\ u_{t-1} \end{array} \right) \right)' \\
\left( \begin{array}{c} A'_{t} \\ B'_{t} \end{array} \right) &= \left( \begin{array}{c} A'_{t-1} \\ B'_{t-1} \end{array} \right) + \frac{1}{t} \hat{R}^{-1}_{t-1} x_{t-1} \left( \begin{array}{c} x_{t-1} \\ u_{t-1} \end{array} \right) \left( x_t - \left( \begin{array}{c} A_{t-1} \\ B_{t-1} \end{array} \right) \left( \begin{array}{c} x_{t-1} \\ u_{t-1} \end{array} \right) \right)' \\
\lambda^*_t &= T^{SP}(H_t, A_t, B_t)x_t.
\end{align*}

**Theorem 4 (Asymptotic Optimality of SP-learning)** If LQ.1 - LQ.3 are satisfied then, locally,

\[
(H_t, A_t, B_t) \xrightarrow{a.s.} (H^*, A, B)
\]

\[
F(H_t, A_t, B_t) \xrightarrow{a.s.} F^*
\]

when the recursive algorithm is augmented with a suitable projection facility.

See the Appendix for the proof, including a more careful statement of the Theorem, a construction of the relevant neighborhood, and a discussion of the “projection facility,” which essentially prevents the estimates from wandering too far away from the fixed point. A detailed discussion of real-time learning in general and projection facilities in particular is provided by Marcet and Sargent (1989), Evans and Honkapohja

\textsuperscript{17}As is standard in the learning literature, when analyzing real-time learning, the agent is assumed not to use current data on $\lambda^*_t$ to form current estimates of $H$ as this avoids technical difficulties with the recursive formulation of the estimators. See Marcet and Sargent (1989) for discussion and details.

\textsuperscript{18}In the updating equation for $(A_t, B_t)$ it might be natural to replace $\hat{R}^{-1}_{t-1}$ with $\hat{R}^{-1}_t$. However eliminating the resulting simultaneity would considerably complicate the system when placed in standard stochastic recursive algorithm form. This timing choice makes no difference asymptotically. See Marcet and Sargent (1989) for a brief discussion.
(1998) and Evans and Honkapohja (2001). We conclude that under quite general conditions, our simple notion of boundedly optimal behavior is asymptotically optimal, that is, shadow-price learners learn to optimize.

The principal, striking result of the adaptive learning literature is that boundedly rational agents, who update their forecasting models in natural ways, may learn to forecast optimally. Theorem 4 is complementary to this principal result, and equally striking: boundedly optimal decisions can converge asymptotically to fully optimal decisions. By estimating shadow prices, our agent has converted an infinite-horizon problem into a two-period optimization problem, which, given his beliefs, is comparatively straightforward to solve. The level of sophistication needed for boundedly optimal decision-making appears to be quite natural: our agent understands simple dynamic trade-offs, can solve simultaneous linear equations and can run simple regressions. Remarkably, with this level of sophistication, the agent can learn over his lifetime how to optimize based on a single realization of his decisions and the resulting states.

3.3 Exogenous states

Some state variables are exogenous: their conditional distributions are unaffected by the control choices of the agent. Operationally, the $i^{th}$-component of the state is exogenous if the $i^{th}$-row of the matrix $B$ is identically zero. To make boundedly optimal decisions, the agent must forecast future values of exogenous states, but it is not necessary that he track the corresponding shadow prices: there is no trade-off between the agent’s choice and expected realizations of an exogenous state. In our work above, to make clear the connection between shadow-price learning and the Riccati equation, we have ignored the distinction between exogenous and endogenous states; however, in our examination of Euler equation learning in Section 3.4.2, and for the applications in Sections 4 and 5, it is helpful to leverage the simplicity afforded by conditioning only on endogenous states. In this subsection we show that the analogue to Theorem 3 holds when the agents restricts his shadow-price PLM to include only endogenous states as regressors.

Assume that the first $n_1$ entries of the of the state vector $x$ are exogenous: they do not depend on the control choice $u$. We show in the Appendix that, without loss of generality, we can assume $m = n - n_1 = n_2$: the number of controls is equal to the number of endogenous states. For the remainder of this subsection, we use the notation

$$X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}$$

to emphasize the block decomposition of the $n \times n$ matrix $X$, with $X_{11}$ being an $n_1 \times n_1$ matrix, and the remaining matrices conformable. Also, for $n \times m$ matrix $X$,
we use
\[ X = \left( \begin{array}{c} X_1 \\ X_2 \end{array} \right), \quad \text{and} \quad X' = \left( \begin{array}{c} X'_1 \\ X'_2 \end{array} \right) \] (24)
with \( X_1 \) an \( n_1 \times m \) matrix and \( X_2 \) conformable. Using this notation, we may write the matrices capturing the transition dynamics as
\[ A = \left( \begin{array}{cc} A_{11} & 0 \\ 0 & A_{22} \end{array} \right) \quad \text{and} \quad B = \left( \begin{array}{c} 0 \\ B_2 \end{array} \right), \]
where the zeros represent conformable matrices with zeros in all entries, and thus capture the exogeneity of \( x_{1t} \).\(^{19}\)

For simplicity we focus on stylized learning and further assume that the agent knows the transition dynamics \( A \) and \( B \). To make his control decisions, he forecasts the future values of the endogenous states’ shadow prices, which we denote by \( \lambda^*_2 \), using the PLM
\[ \lambda^*_2 = \hat{H}x_t + \text{noise}. \] (25)
Here the hat on the perceived coefficient is used to distinguish the beliefs of our endogenous-state SP learner from those beliefs identified in the previous section. Also, notice that the price of an additional unit of endogenous state may depend on the realization of an exogenous state; hence, we condition \( \lambda^*_2 \) on the entire state vector \( x_t \).

The key observations are the following: if \( \lambda^*_t = (\lambda^*_{1t}, \lambda^*_{2t}) \) is the perceived shadow price of the state vector \( x_t \) in time \( t \) then, because of the block-matrix forms of \( A \) and \( B \) above,

1. \( B'\hat{E}_t\lambda^*_{t+1} = B'_2\hat{E}_t\lambda^*_{2t+1} \)
2. \( (A'\hat{E}_t\lambda^*_{t+1})_2 = A'_{22}\hat{E}_t\lambda^*_{2t+1} \)

Here, the notation \((*)_2\) identifies the last \( n_2 \) rows of the matrix \((*)\), and we remember that the hat on the expectations operator refers to the bounded optimality of our agent. With these observations, we may then use the PLM (25) to revisit the computations following equation (18) and obtain the control choice and the price update of an agent with beliefs \( \hat{H} \):

\[ u_t = (2Q - \beta B'_2\hat{H}B)^{-1}(\beta B'_2\hat{H}A - 2W')x_t \equiv \hat{F}(\hat{H})x_t \] (26)
\[ \lambda^*_2 = \left( -2R_2 - 2W_2\hat{F}(\hat{H}) + \beta A'_{22}\hat{H} \left( A + B\hat{F}(\hat{H}) \right) \right)x_t \equiv \hat{T}^{SP}(\hat{H})x_t. \] (27)

\(^{19}\)We assume that the exogenous states are either stationary or unit root: the eigenvalues of \( A_{11} \) lie within the closed unit disk.
Equations (26) and (27) strongly suggest, and indeed almost prove that our agent will learn to optimize: after all, equation (26) shows that only forecasts of $\lambda_{2t+1}$ are required to make control decisions, and equation (27) implies that to update the shadow prices of the endogenous states, we need forecast only the shadow prices of the endogenous states. To show formally that it is sufficient for asymptotic optimality to condition on the vector of endogenous states, we must work a little harder: we must show that $\hat{T}^{SP}$ has a Lyapunov stable fixed point $\hat{H}^*$; and we must show that, at that fixed point, the agent makes fully optimal decisions, that is, $\hat{F}(\hat{H}^*) = F^*$.

To determine the candidate $\hat{H}^*$, let $P^*$ be the usual solution to the Riccati equation and $H^* = -2P^*$ be the corresponding fixed point of $T^{SP}$. Write

$$H^* = \left( \begin{array}{c} H_1^* \\ H_2^* \end{array} \right),$$

and let $\hat{H}^* = H_2^*$. Noting $B' H^* B = B_2' \hat{H}^* B$, it follows that for $\hat{H}$ near $\hat{H}^*$,

$$\det(2Q - \beta B_2' \hat{H}B) \neq 0,$$

so that $\hat{F}(\hat{H})$ is well-defined. We have the following result:

**Theorem 5** Suppose LQ.1 – LQ.3 are satisfied and let $H^* = -2P^*$ and $\hat{H}^* = H_2^*$. Then

1. $\hat{F}(\hat{H}^*) = F^*$,
2. $\hat{H}^*$ is a fixed point of $\hat{T}^{SP}$,
3. $d\hat{H}/d\tau = \hat{T}^{SP}(\hat{H}) - \hat{H}$ is Lyapunov stable at $\hat{H}$.

Theorem 4 shows that if agents forecast the shadow prices for all states then stylized SP-learning is stable. Theorem 5 demonstrates that, in fact, it is sufficient for agents only to form forecasts of the shadow prices for the endogenous states.

### 3.4 Bounded optimality: alternative implementations

Although shadow-price learning provides an appealing way to implement bounded rationality, our discussion in the Introduction suggests that there may be alternatives. In this section we show that our basic approach can easily be extended to encompass two closely related alternatives: value-function learning and Euler-equation learning. Shadow-price learning focuses attention on the marginal value of the state, as this is the information required to chose controls, and under SP-learning the agent estimates the shadow prices directly. In contrast, value-function learning infers the shadow
prices from an estimate of the value function itself. A second alternative, when the endogenous states exhibit no lagged dependence, is for the agent to estimate shadow prices simply using marginal returns \( r_x \), leading to Euler-equation learning. We consider these two alternatives in turn.

### 3.4.1 Value-function learning

In this implementation we start with Bellman’s equation

\[
V(x_t) = \max_{u_t} r(x_t, u_t) + \beta E_t V(g(x_t, u_t, \varepsilon_{t+1}))
\]  

and the necessary conditions for an optimum

\[
r_u(x_t, u_t) + \beta E_t V_x (g(x_t, u_t, \varepsilon_{t+1})) g_u(x_t, u_t, \varepsilon_{t+1}) = 0.
\]  

Here \( V_x (g(x_t, u_t, \varepsilon_{t+1})) \) denotes the gradient of \( V \) evaluated at \( g(x_t, u_t, \varepsilon_{t+1}) \). SP-learning replaces \( V_x \) with a vector of perceived shadow prices and uses (29) as the basis for a decision rule by solving it for \( u_t \) in terms of \( x_t \). Under value-function learning, agents again use (29) as the basis for a decision rule, but compute \( V_x \) from an estimate of \( V \).

We proceed to the details using our Linear-Quadratic (LQ) framework. Equation (29) then becomes

\[
-2Q u_t - 2W' x_t + \beta B' E_t V_x (Ax_t + Bu_t + C\varepsilon_{t+1})' = 0.
\]  

Recall that under optimal decision-making in the LQ set-up, \( V(x) = -x' P^* x + \zeta \) for the appropriate constant \( \zeta \) given in equation (8). Consistent with this we assume that the agent believes that the value of being in state \( x \) depends quadratically on \( x \). To capture this belief with a perceived law of motion, let \( z(x) = (x_i x_j)_{1 \leq i \leq j \leq n} \) be the vector of relevant variables so that the PLM may be written \( V(x) = q' z(x) \). We track the dependence of \( V \) on \( x \) because we will need to differentiate \( V \) with respect to \( x \) to determine how our agent makes decisions. We also remark that typically \( x_1 = 1 \), so that \( z \) includes a constant and all linear terms \( x_i \).

To make his decision the agent computes \( V_x \) from his beliefs \( V(x) = q' z \) and solves (30) for \( u_t \) given \( x_t \). This choice of \( u_t \) can then be used to update his estimate \( \hat{V}_t \) of \( V(x_t) \) using Bellman’s equation:

\[
\hat{V}_t = r(x_t, u_t) + \beta E_t V(Ax_t + Bu_t + C\varepsilon_{t+1}).
\]  

The LQ nature of the functions \( r \) and \( V \) imply that \( \hat{V}_t = T^{VF}(q)' z_t \), for appropriate vector \( T^{VF}(q) \). The corresponding map \( T^{VF} \) is given explicitly in the Appendix as equation (70).
As before, we can distinguish between stylized learning and real-time learning. Here we focus on stylized learning, which, as we saw in Section 3.2.2 on SP-learning, governs the stability of real-time learning through the theory of stochastic recursive algorithms. The value-function learning analog to Theorem 3 is:

**Theorem 6** Assume LQ.1 - LQ.3 are satisfied. Then there exists a Lyapunov stable fixed point \( q^* \) of the differential equation \( dq/d\tau = T^V F(q) - q \) such that \( V(x) = q^* \cdot z \).

We now briefly discuss the real-time algorithm that corresponds to value-function learning. Our agent is assumed to use available data to estimate the transition equation and his value-function PLM \( V(x) = q^' \cdot z(x) \), and then use these estimates to form forecasts and thereby generate new data. The transition equation is estimated just as in Section 3.2.2. To estimate the value-function PLM at time \( t \), and thus obtain an estimate of the beliefs coefficients \( q_t \), we assume our agent regresses \( \hat{V}_{t-1} \) on \( z(x_{t-1}) \) using data \( \{x_{t-1}, \ldots, x_0, \hat{V}_{t-1}, \ldots, \hat{V}_0\} \). Given \( q_t \), the agent’s control choice is

\[
    u_t = F(H(q_t), A_t, B_t) x_t,
\]

where \( H(q_t) = -2P(q_t) \) and \( P(q_t) \) is the \( n \times n \) matrix such that \( q_t^'z(x) = -x^'P(q_t)x + \zeta \). Finally, the agent’s estimated value at time \( t \) is given by

\[
    \hat{V}_t = T^V F(q_t, A_t, B_t) \cdot z_t,
\]

where we abuse notation somewhat by incorporating into the function \( T^V \) the time-varying estimates of \( A \) and \( B \). A causal recursive dynamic system of the form (23) may then be used to state and prove a convergence theorem analogous to Theorem 4.

### 3.4.2 Euler-equation learning

We now consider Euler equation learning. It is revealing to begin with the general set-up. Recall the Bellman system (28) associated with the standard dynamic programming problem (1)-(2). The first-order and envelope conditions are

\[
    0 = r_u(x_t, u_t)' + \beta E_t g_u(x_t, u_t, \epsilon_t+1)' V_x(x_{t+1})' \tag{31}
\]

\[
    V_x(x_t)' = r_x(x_t, u_t)' + \beta E_t g_x(x_t, u_t, \epsilon_t+1)' V_x(x_{t+1})'. \tag{32}
\]

Stepping (32) ahead one period and inserting into (31) yields

\[
    0 = r_u(x_t, u_t)' + \beta E_t (g_u(x_t, u_t, \epsilon_{t+1})' (r_x(x_{t+1}, u_{t+1})' + \beta g_x(x_{t+1}, u_{t+1}, \epsilon_{t+2})' V_x(x_{t+2})')).
\]

Observe that if

\[
    E_t g_u(x_t, u_t, \epsilon_{t+1})' g_x(x_{t+1}, u_{t+1}, \epsilon_{t+2})' V_x(x_{t+2})' = 0 \tag{33}
\]
then we obtain the usual Euler equation

$$0 = r_u(x_t, u_t) + \beta E_t g_u(x_t, u_t, \varepsilon_{t+1})' r_x(x_{t+1}, u_{t+1})'.$$

(34)

The condition (33) is met when the transition equation does not exhibit dependence on endogenous state variables. Specifically, assume $x_{1t+1} = g^1(x_{1t}, \varepsilon_{t+1})$, that is, $x_{1t}$ is the exogenous component of the state $x_t$. Further, assume that the endogenous component, $x_{2t}$, has transition given by $x_{2t+1} = g^2(u_t, \varepsilon_{t+1})$. It follows that

$$g_u(x_t, u_t, \varepsilon_{t+1})' g_x(x_{t+1}, u_{t+1}, \varepsilon_{t+2})' = \left( \begin{array}{c} 0 \\ g_u^2(u_t, \varepsilon_{t+1}) \end{array} \right)' \left( \begin{array}{c} 0 \\ g_{x1}^1(x_{1t+1}, \varepsilon_{t+2}) \end{array} \right) = 0,$n

and thus condition (33) is satisfied.

We return to the LQ set-up to discuss the implementation of Euler-equation learning. Adopting the set-up of Section 3.3, we impose the additional restrictions that

$$A = \left( \begin{array}{cc} A_{11} & 0 \\ 0 & 0 \end{array} \right), \quad B = \left( \begin{array}{c} 0 \\ B_{21} \end{array} \right),$$

i.e. the transition equation does not exhibit dependence on endogenous state variables. Recalling that $g_x = A$ and $g_u = B$ equation (34) becomes

$$Q u_t + W' x_t + \beta B' E_t (R x_{t+1} + W u_{t+1}) = 0.$$

(35)

To implement Euler equation learning, we follow Honkapohja, Mitra, and Evans (2013) and take (35) as the behavioral primitive: our agent forms boundedly rational forecasts of $x_{t+1}$ and $u_{t+1}$, and makes his time $t$ control decision to meet his Euler equation. We adopt stylized learning and for simplicity assume the agent knows the transition matrices $A$ and $B$.

The agent is required to forecast his own future control decision, and we provide him a forecasting model that takes the same form as optimal behavior: $u_t = F x_t$. The agent computes

$$\hat{E}_t x_{t+1} = A x_t + B u_t \quad \text{and} \quad \hat{E}_t u_{t+1} = F \hat{E}_t x_{t+1},$$

(36)

which may then be used in conjunction with (35) to determine his control decision. The following notation will be helpful: for “appropriate” $n \times n$ matrix $X$, set

$$\Phi(X) = (Q + \beta B' X B)^{-1} \quad \text{and} \quad \Psi(X) = \beta B' X A + W',$$

where $X$ is appropriate provided that $\det(Q + \beta B' X B) \neq 0$. Then, combining (36) with (35) and simplifying yields

$$u_t = T^{EL}(F) x_t, \quad \text{where} \quad T^{EL}(F) = -\Phi(R + W F) \Psi(R + W F).$$

(37)

\footnotetext{20We assume that the exogenous states are either stationary or unit root: the eigenvalues of $A_{11}$ lie within the closed unit disk.}
Equation (37) may be interpreted as the actual law of motion given the agent’s beliefs $F$. Finally, the agent updates his forecast of future behavior by regressing the control on the state. This updating process results in a recursive algorithm analogous to (23), which identifies the agent’s behavior over time.

Let $F^* = -\Phi(P^*)\Psi(P^*)$, where $P^*$ is the solution to the Riccati equation. By Theorem 1, we know that $u = F^*x$ is the optimal feedback rule. Also, because $R + WF^* = P^*$, it follows that $TE^L(F^*) = F^*$. Analogous to our earlier results, whether the agent learns over time to behave optimally is determined by the stability of the matrix differential equation $dF/d\tau = TE^L(F) - F$. In fact we have

**Theorem 7** Assume LQ.1 – LQ.3 are satisfied and let $F^* = -\Phi(P^*)\Psi(P^*)$, where $P^*$ is the solution to the Riccati equation. Then $F^*$ is a Lyapunov stable fixed point of the differential equation $dF/d\tau = TE^L(F) - F$.

We now briefly discuss the real-time algorithm that corresponds to Euler-equation learning. Our agent is assumed to use available data to estimate the transition equation and his decision-rule PLM $u = Fx$, and then uses these estimates to form forecasts and make decisions, thereby generating new data. The transition equation is estimated just as in Section 3.2.2. To estimate the beliefs coefficients $F$, we assume our agent regresses $u_{t-1}$ on $x_{t-1}$ using data $\{x_{t-1}, \ldots, x_0, u_{t-1}, \ldots, u_0\}$. Given the time $t$ estimate $F_t$, the agent’s control choice is

$$u_t = TE^L(F_t, A_t, B_t) x_t,$$

where again we abuse notation somewhat by incorporating into the function $TE^L$ the time-varying estimates of $A$ and $B$. A causal recursive dynamic system of the form (23) may then be used to state and prove a convergence theorem for Euler-equation learning analogous to Theorem 4.

We were able to obtain the Euler-equation learning procedure used in this section based on the assumption that the transition equation does not exhibit dependence on endogenous state variables. Often it is possible in more general circumstances to derive a multi-period Euler equation. We illustrate this point in the context of the example given in Section 4.

**3.5 Summary**

This section, which presents our main results, provides theoretical justification for a class of boundedly rational and boundedly optimal decision rules, based on adaptive learning, in which an agent facing a dynamic stochastic optimization problem makes decisions at time $t$ to meet his perceived optimality conditions, given his beliefs about the values of an extra unit of the state variables in the coming period and his perceived
trade-off between controls and states between this period and the next. We fully develop the approach in the context of shadow-price learning in which our agent uses natural statistical procedure to update each period his estimates of the shadow prices of states and of the transition dynamics. Our results show that in the standard Linear-Quadratic setting, an agent following our decision-making and updating rules will make choices that converge over time to fully optimal decision-making. These convergence results extend to alternative variations based on value-function learning and Euler-equation learning. Our results are the bounded optimality counterpart of the now well-established literature on the convergence of least-squares learning to rational expectations. In the remaining sections we show how to apply our results to several standard economic examples.

4 SP-learning in a Crusoe economy

By Theorem 4, an individual with quadratic preferences and facing a linear constraint can learn to make optimal choices provided he makes boundedly rational forecasts and uses boundedly optimal behavior. To illustrate our results, we turn to a simple Crusoe environment with quadratic preferences and linear production specification.

4.1 A Robinson Crusoe economy

A narrative approach may facilitate intuition. Thus, imagine Robinson Crusoe, a middle class Brit, finding himself marooned on a tropical island. An organized young man, he quickly takes stock of his surroundings. He finds that he faces the following problem:

\[
\begin{align*}
\max \quad & -E \sum_{t \geq 0} \beta^t \left((c_t - b_t)^2 + \phi l_t^2\right) \\
\text{s.t.} \quad & y_t = A_1 s_t + A_2 s_{t-1} + z_t \\
& s_{t+1} = y_t - c_t + \mu_{t+1} \\
& s_t = l_t \\
& b_t = b^* + \Delta(b_{t-1} - b^*) + \varepsilon_t \\
& z_t = \rho z_{t-1} + \eta_t, \\
& \text{with } s_{-1}, s_0, \text{ and } z_0, b_0 \text{ given.}
\end{align*}
\]

Here \(y_t\) is fruit and \(c_t\) is consumption of fruit. Equation (39) is Bob’s production function – he can either plant the fruit or eat it, seeds and all – and the double lag captures the production differences between young and old fruit trees. All non-consumed seeds are planted. Some seasons, wind brings in additional seeds from nearby islands; other seasons, local voles eat some of the seeds: thus \(s_{t+1}\), the number
of young trees in time \( t+1 \), is given by equation (40), where the white noise term \( \mu_{t+1} \) captures the variation due to wind and voles. Note that \( s_t \) is both the quantity of young trees in \( t \) and the quantity of old trees in \( t+1 \). Weeds are prevalent on the island: without weeding around all the young trees, the weeds rapidly spread everywhere and there is no production at all from any trees: see equation (41). This is bad news for Bob as he’s not fond of work: \( \phi > 0 \). Finally, \( z_t \) is a productivity shock (rabbits eat saplings and ancient seeds sprout) and \( b_t \) is stochastic bliss: see Ch. 5 of Hansen and Sargent (2014) for further discussion of this economy as well as many other examples of economies governed by quadratic objectives and linear transitions.\(^{21}\)

Some comments on \( \phi \) and the constraint (41) are warranted, as they play important roles in our analysis. Because \( \phi > 0 \) and \( l_t = s_t \), it follows that an increased stock of productive trees reduces Bob’s utility. In the language of LQ programming, these assumptions imply that a diverging state \( s_t \) is observed and must be avoided: specifically, \( \phi > 0 \) is necessary for the corresponding matrix pairs to be observable: see Assumption LQ.3. In contrast, and somewhat improbably, Bob’s disheveled American cousin Slob is not at all lazy: his \( \phi \) is zero and his behavior, which we analyze in a companion paper, is quite different from British Bob’s.

When he is first marooned, Bob does not know if there is a cyclic weather pattern; but he thinks that if last year was dry this year might be dry as well. Good with numbers, Bob decides to estimate this possible correlation using RLS. Bob also estimates the production function using RLS. Finally, Bob contemplates how much fruit to eat. He decides that his consumption choice should depend on the value of future fruit trees forgone. He concludes that the value of an additional tree tomorrow will depend (linearly) on how many trees there are, and makes a reasoned guess about this dependence. Given this guess, Bob estimates the value of an additional tree tomorrow, and chooses how much fruit to eat today.

Belly full, Bob pauses to reflect on his decisions. Bob realizes his consumption choice depended in part on his estimate about the value of additional trees tomorrow, and that perhaps he should revisit this estimate. He decides that the best way to do this is to contemplate the value of an additional tree today. Bob realizes that an additional young tree today requires weeding, but also provides additional young trees tomorrow (if he planted the young tree’s fruit) and an old tree tomorrow, and that an additional old tree today provides young trees tomorrow (if he planted the old tree’s fruit). Using his estimate of the value of additional trees tomorrow, Bob estimates the value of an additional young tree and an additional old tree today. He then uses these estimates to re-evaluate his guess about the dependence of tree-value

\(^{21}\)The only novelty in our economy is the presence of a double lag in production. The double lag is a mechanism to expose the difference between Euler equation learning and SP-learning. Other mechanisms, such as the incorporation of habit persistence in the quadratic objective, yield similar results.
on tree-stock. Exhausted by his efforts, Bob falls sound asleep. He should sleep well: Theorem 4 tells us that by following this simple procedure, Bob will learn to optimally exploit his island paradise.

This simple narrative describes the behavior of our boundedly optimal agent. It also points to a subtle behavioral assumption that is more easily examined by adding precision to the narrative. To avoid unnecessary complication, set $\Delta = 0, z_t = 0, \varepsilon_t = 0$. The simplified problem becomes

$$\max_{t \geq 0} -\hat{E} \sum \beta^t \left((c_t - b^*)^2 + \phi s_t^2\right)$$

subject to

$$s_{t+1} = A_1 s_t + A_2 s_{t-1} - c_t + \mu_{t+1}$$

In notation of Section 3, the state vector is $x_t = (1, s_t, s_{t-1})'$ and the control is $u_t = c_t$. We assume that $\beta A_1 + \beta^2 A_2 > 1$ and $\phi > 0$ to guarantee that steady-state consumption is positive and below bliss: see the Appendix for a detailed analysis of the steady state and fully optimal solution to (42). For the reasons given in Section 3.3 there is no need to forecast the shadow price of the intercept. Thus let $\lambda^*_t$ be the time $t$ value of an additional new tree in time $t$ and $\lambda^*_{2t}$ the time $t$ value of an additional old tree in time $t$. Bob guesses that $\lambda^*_t$ depends on $s_t$ and $s_{t-1}$:

$$\lambda^*_t = a_i + b_i s_i + d_i s_{t-1}, \text{ for } i = 1, 2. \quad (43)$$

He then forecasts $\lambda^*_{t+1}$:

$$\hat{E}_t \lambda^*_{t+1} = a_i + b_i \hat{E}_t s_{t+1} + d_i s_t, \text{ for } i = 1, 2. \quad (44)$$

Because he must choose consumption, and therefore savings, before output is realized, Bob estimates the production function and finds

$$\hat{E}_t s_{t+1} = A_{1t} s_t + A_{2t} s_{t-1} - c_t,$$

where $A_{it}$ is obtained by regressing $s_t$ on $(s_{t-1}, s_{t-2})'$. He concludes that

$$\hat{E}_t \lambda^*_{it+1} = a_i + (b_i A_{1t} + d_i) s_t + b_i A_{2t} s_{t-1} - b_i c_t, \quad (45)$$

which, he notes, depends on his consumption choice today.

Now Bob contemplates his consumption decision. By increasing consumption by $dc$, Bob gains $-2(c_t - b^*)dc$ and loses $\beta E_t \lambda^*_{t+1} dc$. Bob equates marginal gain with marginal loss, and obtains

$$c_t = b^* - \frac{\beta}{2} E_t \lambda^*_{t+1}. \quad (46)$$

This equation in conjunction with (45) allows Bob to obtain numerical values for his consumption and forecasted shadow prices.\textsuperscript{22}

\textsuperscript{22}See the Appendix for formal details linking this example to the set-up of Section 3.
Finally, Bob revisits his parameter guesses \( a_i, b_i, \) and \( d_i \). He first thinks about the benefit of an additional new tree today: it would require weeding, but the fruits could be saved to produce \( A_{1t} \) new trees tomorrow, plus he gets an additional old tree tomorrow. He concludes

\[
\lambda^*_{1t} = -2\phi s_t + \beta A_{1t} E_t \lambda^*_{1t+1} + \beta E_t \lambda^*_{2t+1}. \tag{47}
\]

He then thinks about the benefit of an additional old tree today: the fruits could be saved to produce \( A_{2t} \) new trees tomorrow. Thus

\[
\lambda^*_{2t} = \beta A_{2t} E_t \lambda^*_{1t+1}. \tag{48}
\]

Because Bob has numerical values for \( E_t \lambda^*_{1t+1} \), (47) and (48), together with the estimates \( A_{it} \), generate numerical values for the perceived shadow prices. Bob will then use these data to form new estimates of his parameter guesses \( a_i, b_i, \) and \( d_i \). We have the following result.

**Proposition 8** *Robinson Crusoe learns to optimally consume fruit.*

The proof of this proposition is in the Appendix.

This implementation of the narrative above highlights our view of Bob’s behavior: he estimates forecasting models, makes decisions, and collects new data to update his models. Thus Bob understands simple trade-offs, can solve simultaneous linear equations and can run simple regressions. These skills are the minimal requirements for boundedly optimal decision-making in a dynamic, stochastic environment. Remarkably, they are also sufficient for asymptotic optimality.

One might ask whether Bob should be more sophisticated. For example Bob might search for a forecasting model that is consistent with the way shadow prices are subsequently revised: Bob could seek a fixed point of the \( T^{SP} \)-map. We view this alternative behavioral assumption as too strong, and somewhat unnatural, for two reasons. First, we doubt that in practice most boundedly rational agents explicitly understand the existence of a \( T^{SP} \)-map. Even if an agent did knew the form of the \( T^{SP} \)-map, would he recognize that a fixed point is what is wanted to ensure optimal behavior? Why would the agent think such a fixed point even exists? And if it did exist, how would the agent find it? Recognition that a fixed point is important, exists, and is computable is precisely the knowledge afforded those who study dynamic programming; our assumption is that our agent does not have this knowledge, even implicitly.

Our second reason for assuming Bob does not seek a fixed point to the \( T^{SP} \)-map relates to the above observation that obtaining such a fixed point is equivalent to full optimality given the “perceived transition equations.” However, if the perceived coefficients \( A_{it} \) are far from the true coefficients, \( A_t \), it is not clear that the behavior
dictated by a fixed point to the $T^{SP}$-map is superior to the behavior we assume. Given that computation is unambiguously costly, it makes more sense to us that Bob not iterate on the $T^{SP}$-map for fear that he might make choices based on magnified errors.

The central point of our paper is precisely that with limited sophistication, plausible and natural boundedly optimal decision-making rules converge to fully optimal decision-making.

### 4.2 Comparing learning mechanisms in a Crusoe economy

The simplified model (42) provides a nice laboratory to compare and contrast SP-learning with Euler equation learning. For simplicity we adopt stylized learning and assume that our agent knows the true values of $A_t$. Shadow price learning has been detailed in the previous section: the agent has PLM (43), and using this PLM, he forecasts future shadow prices: see (44). These forecasts yield his consumption decision

$$c_t = \phi_1(a_i, b_i, d_i) + \phi_3(a_i, b_i, d_i)s_t + \phi_3(a_i, b_i, d_i)s_{t-1},$$

which he uses to compute shadow price forecasts via (45). We may use these forecasts, together with equations (47) and (48) to determine $T^{SP}$-map: see the Appendix for details.

The conditions given in Section 3.4.2 for obtaining a first-order Euler equation are not satisfied for the current example. However, it is possible to obtain the Euler equation

$$c_t - \beta\phi E_t s_{t+1} = \Psi + \beta A_1 E_t c_{t+1} + \beta^2 A_2 E_t c_{t+2},$$

where $\Psi = b^*(1 - \beta A_1 - \beta^2 A_2)$, which is a second-order Euler equation when $A_2 \neq 0$. We can then proceed analogously to Section 3.4.2 and implement Euler equation learning by taking (50) as a behavioral primitive: the agent is assumed to forecast his future consumption behavior and then choose consumption today based on these forecasts. The agent is assumed to form forecasts using a PLM that is functionally consistent with optimal behavior:

$$c_t = a_3 + b_3s_t + d_3s_{t-1}.$$

Using this forecasting model and the transition equation

$$s_{t+1} = A_1s_t + A_2s_{t-1} - c_t + \mu_{t+1},$$

the agent chooses $c_t$ to satisfy (50). This behavior can then be used to identify the associated T-map. See the Appendix for a derivation of the T-map.

---

23 The Euler equation can be derived by a direct variational argument. Alternatively it can be obtained from (46), (47) and (48).
Our interest here is to compare shadow price learning and Euler equation learning. Although we have not explored this point in the current paper, it is known from the literature on stochastic recursive algorithms that the speed of convergence of the real-time versions of our set-ups is governed by the maximum real parts of the eigenvalues of the T-map’s derivative: this maximum needs to be less than one for stability and larger values lead to slower convergence. Here, if \( A_2 = 0 \) then the agent’s problem has a one dimensional control and a two dimensional state, with one dimension corresponding to a constant: in this case the Euler equation is first-order and shadow price learning and Euler equation learning are equivalent. However, for \( A_2 > 0 \) the endogenous state’s dimension is two and the equivalence may break down, as is evidenced by Figure 1, which plots the maximum real part of the eigenvalues for the respective T-map’s derivatives.

![Figure 1: SP learning vs Euler learning: largest eigenvalue](image)

The intuition for the inequivalence of shadow-price learning and Euler equation learning is straightforward: shadow price learning recognizes the two endogenous states and estimates the corresponding PLM. In contrast, under Euler equation learning agents need to understand and combine several structure relationships: they must understand the relationship between the two shadow prices;\(^{24}\) and, they must combine this understanding with the first-order condition for the controls to eliminate the dependence on these shadow prices. In our simple example, Bob, as a shadow-price learner, would need to combine equations (47) and (48) with his decision rule to obtain the Euler equation (50). In this sense, shadow-price learning requires less

\(^{24}\)We note that for the general LQ-problem, the relationship between shadow prices may be quite complicated.
structural information than Euler equation learning.

5 SP Learning in a Ramsey Model

Our principal result on SP learning formally establishes that a quadratic regulator facing a linear constraint may, by estimating the transition equation and making control decisions conditional on forecasts of the states’ shadow prices, learn to behave optimally. The linear/quadratic environment is key to obtaining analytic results – the resulting stochastic system is linear – and it is arguably natural in some economic settings; however, most DSGE frameworks embrace a more general stochastic decision-making model. As emphasized in Section 2.2, while the theoretical arguments are not known to hold in the general settings, the intuition on which SP learning is built – that agents make decisions contingent on perceived values of states – still has merit and its implementation can be explored computationally. Here, we consider a Ramsey model as a simple example of SP-learning in a non-LQ environment.

5.1 The model and the REE

The modeling environment is entirely standard, and we will present it only briefly. Labor is supplied inelastically. The representative agent’s problem is given by

$$\max \ E \sum \beta^t U(c_t) \quad s_t = (1 + r_t)s_{t-1} + w_t - c_t.$$ 

Here, the endogenous state for the agent at time $t$ is $s_{t-1}$ and the agent’s control at time $t$ is $c_t$. Firms own CRTS technology, which is given by $y = zf(k)$ in per-worker units. Here, $\log z$ is a stationary AR(1) stochastic productivity shock with small support, namely $z_t = \varepsilon_t z_{t-1}^\rho$, where $\varepsilon_t$ is iid exogenous with mean one. Factor prices are assigned their marginal products.

The rational expectations equilibrium is the unique bounded path satisfying

$$U'(c_t) = \beta E_t(1 + z_{t+1}f'(k_{t+1}) - \delta)U'(c_{t+1})$$

$$k_{t+1} = z_t f(k_t) + (1 - \delta)k_t - c_t$$

$$z_t = \varepsilon_t z_{t-1}^\rho.$$ 

Near the non-stochastic steady state $(k, c)$, the REE is approximated by

$$c_t = c^* + Ak_t + Bz_t$$

$$k_{t+1} = k^* + \psi k_t - c_t + f(k)z_t$$

$$z_t = -\rho + \rho z_{t-1} + \varepsilon_t$$
for $\psi = f'(k) + 1 - \delta$ and appropriate $A, B, c^*$ and $k^*$. Notice that we are representing the REE in levels.

Finally, note that if $U(c) = \log(c)$, $f(k) = k^\alpha$, and $\delta = 1$, then we have that

$$c_t = (1 - \alpha \beta) z_t k_t^\alpha \quad \text{and} \quad k_{t+1} = \alpha \beta z_t k_t^\alpha.$$ 

This formulation will be handy when assessing the quality of our boundedly rational agent’s behavior.

5.2 Bounded optimality

To study bounded-rationality/optimality, it is common at this point (indeed ubiquitous) to linearize the appropriate equations (Euler equations, perhaps combined with LTBC, capital accumulation, and shocks), and only then impose, for example, adaptive learning behavior. We could do exactly this, and we would find that, for the model under examination, the behavior under SP-learning is identical to that implied by Euler-equation learning. Instead, however, for this exercise we continue to take as literal the non-linear economy, and adapt our agent’s behavioral primitives accordingly.25

The agent’s decision problem is given by

$$\max \quad E_0 \sum \beta^t U(c_t)$$

subject to

$$s_{t+1} = (1 + r_t) s_t + w_t - c_t.$$ 

The agent treats $w_t$ and $r_t$ as exogenous, and in line with the literature we assume that $z_t$ is observed and known by agents to be a determinant of factor prices.

We put the agent’s decision problem in the general form (1)-(2) as follows. The control is $c_t$ and the state is $x_t = (1, s_t, w_t, r_t, z_t)$, with transition function $g(x_t, c_t, \varepsilon_{t+1})$, where

$$g(x_t, c_t, \varepsilon_{t+1}) = \begin{pmatrix} 1 \\ (1 + r_t) s_t + w_t - c_t \\ g^w(x_t, c_t, \varepsilon_{t+1}) \\ g^r(x_t, c_t, \varepsilon_{t+1}) \\ \varepsilon_{t+1} \varepsilon_t^\nu \end{pmatrix}.$$ 

The agent treats $w_t, r_t$ and $z_t$ as exogenous. From Theorem 5 the only relevant shadow price for decision-making measures the value of the variable $s_t$. Under optimal decision-making the shadow price of $s_t$ depends in a non-linear way on the state vector $x_t$. Instead of assuming that the agent understands and knows the form

$$25 \text{It would be straightforward to allow the agent to estimate the } z_t \text{ process and this would not change our results.}$$

36
of this non-linear dependency, we follow the discussion of Section 2.2 and assume that the agent uses a linear model to forecast the shadow price. This is a plausible misspecification consistent with the view that, in general environments involving uncertainty, agents are unlikely to know the true data-generating process and will thus approximate its functional form. Using linear forecasting models is one particularly natural approximation; other common approximations include parsimonious forecasting models that economize on the number of independent variables and their lags.

With this discussion in mind, the PLM for the shadow price, as in Section 2.2, is $\lambda_t^* = H x_t$. However, in our representative-agent framework including all of the state components as regressors would lead to severe multicollinearity problems because $\omega_t$ and $\rho_t$ are exact functions of $k_t$ and $z_t$, and in equilibrium $s_t = k_t$. To deal with this problem, we need to reduce by two the number of regressors, and for convenience we choose the regressors to be $\tilde{x}_t = (1, s_t, z_t)$.

As in Section 2.2, the agent needs to forecast $\tilde{x}_{t+1}$, and in the current setting this is straightforward. Because (52) is the agent’s flow budget constraint, we naturally assume that this equation is known by the agent and used to perfectly forecast $s_{t+1}$. For simplicity we also assume that the linearized transition for productivity is known so that $\hat{E}_t z_{t+1} = -\rho + \rho z_t$. Thus

$$
\hat{E}_t \lambda_{t+1}^* = H \hat{E}_t \tilde{x}_{t+1} = H_s s_{t+1} + H_z \hat{E}_t \tilde{z}_{t+1},
$$

i.e.

$$
\hat{E}_t \lambda_{t+1}^* = H - \rho H_z + H_s ((1 + r_t) s_t + w_t) + \rho H_z z_t - H_s c_t.
$$

The agent’s decision and perceived shadow price, via the analogs to (5) and (6), are given by

$$
U'(c_t) = \beta \hat{E}_t \lambda_{t+1}^*,
$$

$$
\lambda_t^* = \beta (1 + r_t) \hat{E}_t \lambda_{t+1}^*,
$$

where we remark that $\lambda_t^*$ satisfies $\lambda_t^* = (1 + r_t) U'(c_t)$. Combining the equation for $\hat{E}_t \lambda_{t+1}^*$ with $U'(c_t) = \beta \hat{E}_t \lambda_{t+1}^*$ we obtain $c_t = c(s_t, w_t, r_t, z_t, H)$.

The beliefs parameters $H$ are determined using OLS. Thus $H_t$, the time $t$ estimate of $H$, is obtained by regressing $\lambda_t^*$ on $\tilde{x}_t$ using data $\{\tilde{x}_0, ..., \tilde{x}_{t-1}\}$ and $\{\lambda_0^*, ..., \lambda_{t-1}^*\}$.26

26The estimate $H_t$ is determined using time $t-1$ data because computation of $\lambda_t^*$ requires $H_t$.  

37
A recursive causal system determining the evolution of the economy is given by

\[
\begin{align*}
\mathcal{R}_t &= \mathcal{R}_{t-1} + \gamma_t (\bar{x}_{t-1} \bar{x}'_{t-1} - \mathcal{R}_{t-1}) \\
H_t &= H_{t-1} + \gamma_t \mathcal{R}_t^{-1} \bar{x}_{t-1} (\lambda_t^* - H'_{t-1} \bar{x}_{t-1}) \\
z_t &= z_{t-1} \\
w_t &= z_t (f(k_t) - f'(k_t) k_t) \\
r_t &= z_t f'(k_t) - \delta \\
c_t &= c(s_t, r_t, w_t, z_t, H_t) \\
s_{t+1} &= (1 + r_t) s_t + w_t - c_t \\
\lambda_t^* &= (1 + r_t) U'(c_t) \\
k_{t+1} &= s_{t+1}.
\end{align*}
\]

To simulate the model, an initial condition is needed. In a REE the true shadow price satisfies \( \lambda_t = (1 + r_t) U'(c_t) \). Linearizing this dependence of \( \lambda \) on \( z_t, k_t \) and \( c_t \), and imposing the linearized REE dependence of \( c_t \) on \( k_t \) and \( z_t \) yields the linear approximation

\[
\lambda_t = \hat{H}_s + \hat{H}_k k_t + \hat{H}_z z_t.
\]

Because in equilibrium \( s_t = k_t \), \( \hat{H} \) provides a natural benchmark for the initial value of \( H \), and we can also compute the covariance matrix of \( (k_t, z_t) \) in the linearized REE to help determine a benchmark for \( \mathcal{R}_0 \). Initial conditions are then chosen as perturbations of these values. We note that we should not expect the beliefs \( H \) in our model to converge to \( \hat{H} \): indeed, while our agents are using linearized forecasting models, the environment is non-linear and the appropriate equilibrium concept is what is often called a “restricted perceptions equilibrium” in the adaptive learning literature.

### 5.3 An illustration

As an experiment, we set \( \delta = 1, f(k) = k^\alpha \) and \( U(c) = \log(c) \). The analytic REE shadow price is computed to be

\[
\lambda_t = \frac{\alpha}{(1 - \alpha \beta) k_t}.
\]

Note that in this special case there is no dependency of \( \lambda_t \) on \( z_t \).

Given a beliefs vector \( H_t \), the agent holds time \( t \) perceptions \( \lambda_t^* = H'_t \cdot (1, s_t, z_t)' \). To assess the quality of our agent’s asymptotic behavior, we run a simulation and record his initial and final-period beliefs. We then plot his initial and final perceived shadow-price dependence against \( s \), setting \( z = 1 \) (its steady-state value), and compare these plots to each other and to the plot of \( \lambda \) as given by (53): see Figure 2.
In Figure 2, the (red) dashed line is initial perceptions and the (black) solid line is final perceptions. The dashed (blue) curve is the true multiplier in the REE. We note that the long-run perceptions of the agent appear to coincide, to first order, with the true dependence of the (optimal) shadow price on capital. We take this as evidence that even in our non-LQ environment our representative agent’s asymptotic behavior is approximately optimal.

Figure 2: Convergence of beliefs in Ramsey model. Curved dash line gives actual shadow price in REE. Straight dashed line gives initial perceived shadow price. Straight solid line is asymptotic perceived shadow price under learning.

6 Conclusion

The prominent role played by micro-foundations in modern macroeconomic theory has directed researchers to intensely scrutinize the assumption of rationality – an assumption on which these micro-foundations fundamentally rest; and, some researchers have criticized the implied level of sophistication demanded of agents in these micro-founded models is unrealistically high. Rationality on the part of agents consists of two central behavioral primitives: that agents are optimal forecasters; and that agents make optimal decisions given these forecasts. While the macroeconomics learning literature has defended the optimal forecasting ability of agents by showing that agents
may learn the economy’s rational expectations equilibrium, and thereby learn to forecast optimally, the way in which agents make decisions while learning to forecast has been given much less attention.

In this paper, we formalize the connection between boundedly rational forecasts and agents’ choices by introducing the notion of bounded optimality. Our agents follow simple behavioral primitives: they use econometric models to forecast one-period ahead shadow prices; and they make control decisions today based on the trade-off implied by these forecasted prices. We call this learning mechanism shadow-price learning. We find our learning mechanism appealing for a number of reasons: it requires only simple econometric modeling and thus is consistent with the learning literature; it assumes agents make only one-period-ahead forecasts instead of establishing priors over the distributions of all future endogenous variables; and it imposes only that agents make decisions based on these one-period-ahead forecasts, rather than requiring agents to solve a dynamic programming problem with parameter uncertainty.

Investigation of SP-learning reveals that it is behaviorally consistent at the agent level: by following our simple behavioral assumptions, an individual facing a standard dynamic programming problem will learn to optimize. Our central stability results are shown to imply asymptotic optimality of alternative implementations of boundedly optimal decision-making, including value-function and Euler-equation learning. Further, we find that SP-learning embeds naturally in the Ramsey model, and clearly this procedure can be used in more elaborate DSGE settings.

It may appear tempting to take our results to suggest that agents should simply be modeled as fully optimizing. However, we think the implications of our results are far-ranging. Since agents have finite lives, learning dynamics will likely be significant in many settings. If there is occasional structural change, transitional dynamics will also be important; and if agents anticipate repeated structural change, say in the form of randomly switching structural transition dynamics, and if as a result they employ discounted least squares, as in Sargent (1999) and Williams (2014), then there will be perpetual learning that can include important escape dynamics. Finally, as we found in the Ramsey model example of Section 5, agents with plausibly misspecified forecasting models will only approximate optimal decision making. As in the adaptive learning literature, we anticipate that our convergence results will be the leading edge of a family of approaches to boundedly optimal decision making. We intend to explore this family of approaches in future work.
Appendix

Proof of Lemma 2. For notational convenience, let $\Psi(P) = \beta B'PA + W'$ and $\Phi(P) = (Q + \beta B'PB)^{-1}$. Then $T(P) = R + \beta A'PA - \Psi(P)'\Phi(P)\Psi(P)$. Working with matrix differentials, and noting that $d\Phi(P) = -\beta \Phi(P)'B'dPB\Phi(P)$, we have

$$
\begin{align*}
\frac{dT}{dP} &= \beta^{1/2}A'dPA\beta^{1/2} - \beta^{1/2}A'dPB\Phi(P)\Psi(P)\beta^{1/2} \\
& \quad + \beta^{1/2}\Psi(P)'\Phi(P)'B'dPB\Phi(P)\Psi(P)\beta^{1/2} - \beta^{1/2}\Psi(P)'\Phi(P)'B'dPA\beta^{1/2} \\
& = (\beta^{1/2}A' - \beta^{1/2}\Psi(P)'\Phi(P)'B')dP(\beta^{1/2}A - \beta^{1/2}B\Phi(P)\Phi(P)) \\
& = \Omega(P)'dP\Omega(P),
\end{align*}
$$

where $\Omega(P) = \beta^{1/2}A - \beta^{1/2}B(Q + \beta B'PB)^{-1}(\beta B'PA + W')$.

It follows that $vec(dT) = (\Omega(P)' \otimes \Omega(P)')vec(dP)$, or $D(T_v)(vec(P)) = \Omega(P)' \otimes \Omega(P)'$, and thus it suffices to show that the eigenvalues of $\Omega(P^*)$ have modulus less than one.

To compute these eigenvalues, recall that $\hat{A} = \beta^{1/2}(A - BQ^{-1}W')$, and let $B_1 = \beta^{1/2}BQ^{-1/2}$, where $Q^{-1/2}$ is the usual square root of the positive definite matrix $Q$. Now notice that

$$
\begin{align*}
\hat{A} - B_1(I_n + B_1^*P^*B_1)^{-1}B_1P^*\hat{A} \\
& = \beta^{1/2}A - \beta^{1/2}BQ^{-1}W' - \beta^{1/2}B(Q + \beta B'P^*B)^{-1}\beta^{1/2}B'P^*(A - BQ^{-1}W')\beta^{1/2} \\
& = \beta^{1/2}A - \beta^{1/2}B(Q + \beta B'P^*B)^{-1}(Q + \beta B'P^*B)Q^{-1}W' \\
& \quad - \beta^{1/2}B(Q + \beta B'P^*B)^{-1}\beta^{1/2}B'P^*(A - BQ^{-1}W')\beta^{1/2} \\
& = \beta^{1/2}A - \beta^{1/2}B(Q + \beta B'P^*B)^{-1} \\
& \quad \times ((Q + \beta B'P^*B)Q^{-1}W' + \beta B'P^*(A - BQ^{-1}W')) \\
& = \beta^{1/2}A - \beta^{1/2}B(Q + \beta B'P^*B)^{-1}(\beta B'P^*A + W') = \Omega(P^*).
\end{align*}
$$

To complete the proof, we observe that by assumption $(\hat{A}, \hat{D})$ is detectable. Also, by LQ.2, there is a matrix $K$ so that $\hat{A} + B_1Q^{1/2}K$ is stable. Letting $K^* = Q^{1/2}K$, we conclude that $\hat{A} + B_1K^*$ is stable, so that $(\hat{A}, B_1)$ is stabilizable. The result then follows from Theorem 13.5.2 in Lancaster and Rodman (1995), which states that under these conditions, the eigenvalues of $\hat{A} - B_1(I_n + B_1^*P^*B_1)^{-1}B_1^*P^*\hat{A}$ have modulus less than one. ■

Proof of Theorem 3. It is straightforward to show that $T^{SP}(H) = -2T\left(-\frac{H}{2}\right)$. It follows that

$$
T^{SP}_v(vec(H)) = -2T_v\left(vec\left(-\frac{H}{2}\right)\right).
$$
By the chain rule,

\[
D \left( T^S \right) \left( vec(H^*) \right) = D \left( T_v \right) \left( vec \left( -\frac{H^*}{2} \right) \right) = D \left( T_v \right) \left( vec(P^*) \right),
\]

and the result follows from Lemma 2. ■

**Proof of Theorem 4.** The proof involves using the theory of stochastic recursive algorithms to show that the asymptotic behavior of our system is governed by the Lyapunov stability of the differential system

\[
\frac{dH}{d\tau} = T^{SP}(H, A, B) - H.
\]

The proof is then completed by appealing to Theorem 3.

Recall the dynamic system under consideration:

\[
\begin{align*}
x_t &= Ax_{t-1} + B u_{t-1} + C \varepsilon_t \\
R_t &= R_{t-1} + \frac{1}{t} \begin{pmatrix} x_t x'_t - R_{t-1} \end{pmatrix} \\
H'_t &= H'_{t-1} + \frac{1}{t} R_{t-1}^{-1} x_{t-1} \left( \lambda^*_t - H_{t-1} x_{t-1} \right)' \\
\hat{R}_t &= \hat{R}_{t-1} + \frac{1}{t} \begin{pmatrix} x_{t-1} u'_{t-1} - \hat{R}_{t-1} \end{pmatrix}
\end{align*}
\]

(54)

To apply the theory of stochastic recursive algorithms, we must place our system in the following form:

\[
\begin{align*}
A'_t &= \begin{pmatrix} A'_{t-1} \\
B'_{t-1} \end{pmatrix} + \frac{1}{t} \hat{R}_{t-1}^{-1} \begin{pmatrix} x_{t-1} \\
( A_{t-1}, B_{t-1} ) \end{pmatrix} \begin{pmatrix} x'_{t-1} \\
u'_{t-1} \end{pmatrix}' \\
\lambda^*_t &= T^{SP}(H_t, A_t, B_t) x_t.
\end{align*}
\]

Here \( \theta \in \mathbb{R}^M \) for some \( M \). For extensive details on the asymptotic theory of recursive algorithms such as this, see Chapter 6 of Evans and Honkapohja (2001). Note that \( \mathcal{R}_t, \hat{R}_t, A'_t \) and \( B'_t \) are matrices and \( \theta \) is a column vector. Therefore, we identify the space of matrices with \( \mathbb{R}^M \) for appropriate \( M \) in the usual way using the vec operator.

\(^{27}\) Other key references are Ljung (1977) and Marcet and Sargent (1989).
With these new variables we may define \( \phi_t = (\hat{A}^t', \hat{B}^t)' \), and

\[
\theta_t = \begin{pmatrix}
\text{vec}(R_t) \\
\text{vec}(H_t) \\
\text{vec}(\hat{R}_t) \\
\text{vec}(\hat{\phi}_t)
\end{pmatrix}
\text{ and } X_t = \begin{pmatrix}
x_t \\
x_{t-1} \\
0
\end{pmatrix}.
\]

Note that \( \theta \in \mathbb{R}^{n^2} \times \mathbb{R}^{(n+m)^2} \times \mathbb{R}^{n(n+m)} = \mathbb{R}^M \) for \( M = 2n^2 + (n+m)^2 + n(n+m) \). Fixing an estimate \( \theta \), define the matrices \( \mathcal{A}(\theta) \) and \( \mathcal{B}(\theta) \) as follows:

\[
\mathcal{A}(\theta) = \begin{pmatrix}
A + BF(H, \tilde{A}, \tilde{B}) & 0 & 0 \\
\tilde{I}_n & 0 & 0 \\
F(H, \tilde{A}, \tilde{B}) & 0 & 0
\end{pmatrix}
\text{ and } \mathcal{B}(\theta) = \begin{pmatrix}
C \\
0 \\
0
\end{pmatrix},
\]

where \( H, \tilde{A}, \tilde{B} \) are the matrices corresponding to the relevant components of \( \theta \).

We now restrict attention to an open set \( W \subset \mathbb{R}^M \) such that whenever \( \theta \in W \) it follows that \( T^{SP} \) is well-defined, \( R^{-1} \) and \( \hat{R}^{-1} \) exist, and the eigenvalues of \( \mathcal{A}(\theta) \) have modulus strictly less than one. Given \( \theta \in W \), define \( \hat{X}_t(\theta) \) as the stochastic process \( \hat{X}_t(\theta) = \mathcal{A}(\theta)\hat{X}_{t-1}(\theta) + \mathcal{B}(\theta)\varepsilon_t \). Let \( \bar{x}_t(\theta) \) denote the first \( n \) components of \( \hat{X}_t(\theta) \) and \( \bar{u}_{t-1}(\theta) \) denote the last \( m \) components of \( \hat{X}_t(\theta) \). With the restriction on \( W \) the following limits are well defined:

\[
\mathcal{N}(H, \tilde{A}, \tilde{B}) = \lim_{t \to \infty} E\bar{x}_t(\theta)\bar{x}_t(\theta)' \text{ and } \bar{\mathcal{N}}(H, \tilde{A}, \tilde{B}) = \lim_{t \to \infty} E \left( \begin{pmatrix} \bar{x}_t(\theta) \\ \bar{u}_t(\theta) \end{pmatrix} \right) \left( \begin{pmatrix} \bar{x}_t(\theta)' \\ \bar{u}_t(\theta)' \end{pmatrix} \right).
\]

Set

\[
\theta^* = \begin{pmatrix}
\text{vec}(\mathcal{N}(H^*, A, B)) \\
\text{vec}(H^*) \\
\text{vec}(\bar{\mathcal{N}}(H^*, A, B)) \\
\text{vec}((A', B')')
\end{pmatrix},
\]

where clearly \( \theta^* \in W \).

We now write the recursion (54) in the form (55)-(56). To this end, we define the function \( \mathcal{H}(\cdot, X) : W \to \mathbb{R}^M \) component-wise as follows:

\[
\mathcal{H}^1(\theta_{t-1}, X_t) = \text{vec}(x_t x_t' - R_{t-1}) \\
\mathcal{H}^2(\theta_{t-1}, X_t) = \text{vec}(R_{t-1}^{-1} x_{t-1 - 1} ((T^{SP}(H_{t-1}, A_{t-1}, B_{t-1}) - H_{t-1}) x_{t-1}))' \\
\mathcal{H}^3(\theta_{t-1}, X_t) = \text{vec} \left( \begin{pmatrix} x_{t-1} \\ u_{t-1} \end{pmatrix} \begin{pmatrix} x_{t-1}' \\ u_{t-1}' \end{pmatrix} - \hat{R}_{t-1} \right) \\
\mathcal{H}^4(\theta_{t-1}, X_t) = \text{vec} \left( \hat{R}_{t-1}^{-1} \begin{pmatrix} x_{t-1} \\ u_{t-1} \end{pmatrix} \begin{pmatrix} x_t - (A_{t-1}, B_{t-1}) (x_{t-1})' \end{pmatrix} \right)'.
\]

43
The theory of stochastic recursive algorithms tells us to consider the function \( h : W \rightarrow \mathbb{R}^M \) defined by

\[
h(\theta) = \lim_{t \to \infty} E\mathcal{H}(\theta, \tilde{X}_t(\theta)),
\]

where existence of this limit is guaranteed by our restrictions on \( W \). The function \( h \) has components

\[
\begin{align*}
h^1(\theta) &= \text{vec}(\mathcal{N}(\theta) - \mathcal{R}) \\
h^2(\theta) &= \text{vec}\left( \mathcal{R}^{-1}\mathcal{N}(\theta) \left( T^{SP}(H, \tilde{A}, \tilde{B}) - H \right)' \right) \\
h^3(\theta) &= \text{vec}\left( \hat{\mathcal{N}}(\theta) - \hat{\mathcal{R}} \right) \\
h^4(\theta) &= \text{vec}\left( \hat{\mathcal{R}}^{-1}\hat{\mathcal{N}}(\theta) \left( \begin{pmatrix} A' \\ B' \end{pmatrix} - \begin{pmatrix} \tilde{A}' \\ \tilde{B}' \end{pmatrix} \right) \right),
\end{align*}
\]

and captures the long-run expected behavior of \( \mathcal{H} \).

We will apply Theorem 4 of Ljung (1977), which directs attention to the ordinary differential equation \( \dot{\mathcal{H}} = h(\theta) \), i.e. \( d\theta/d\tau = h(\theta) \), where \( \tau \) denotes notational time. Notice that \( h(\theta^*) = 0 \), so that \( \theta^* \) is a fixed point of this differential equation. Ljung’s theorem tell us that, under certain conditions that we will verify, if \( \theta^* \) is a Lyapunov stable fixed point, then our learning algorithm will converge to it almost surely. The determination of Lyapunov stability for the system \( \dot{\theta} = h(\theta) \) involves simply computing the derivative of \( h \) and studying its eigenvalues: if the real parts of these eigenvalues are negative then the fixed point is Lyapunov stable. Computation of the derivative of \( h \) at \( \theta^* \) is accomplished by observing that the terms multiplying \( \mathcal{N}(\theta) \) and \( \hat{\mathcal{N}}(\theta) \) in equation (58) and (60) are zero when evaluated at \( \theta^* \) so that, by the product rule, the associated derivatives are zero. The resulting block diagonal form of the derivative of \( h \) yields repeated eigenvalues that are \(-1\) and the eigenvalues of \( \partial h^2/\partial \text{vec}(H') \), which have negative real part by Theorem 3. It follows that \( \theta^* \) is a Lyapunov stable fixed point of \( \dot{\theta} = h(\theta) \).

To complete the proof of Theorem 4, we must verify the conditions of Ljung’s Theorem and augment the algorithm (54) with a projection facility. First we address the regularity conditions on the algorithm. Because \( \varepsilon_t \) has compact support, we apply Theorem 4 of Ljung (1977) using his assumptions A. Let the set \( D \) be the intersection of \( W \) with the basin of attraction of \( \theta^* \) under the dynamics \( \dot{\theta} = h(\theta) \). Note that \( D \) is both open and path-connected. Let \( D_R \) be a bounded open connected subset of \( D \) containing \( \theta^* \) such that its closure is also in \( D \). We note that for fixed \( X, \mathcal{H}(\theta, X) \) is continuously differentiable (with respect to \( \theta \)) on \( D_R \), and for fixed \( \theta \in D_R, \mathcal{H}(\theta, X) \) is continuously differentiable with respect to \( X \). Furthermore, on \( D_R \) the matrix functions \( \mathcal{A}(\theta) \) and \( \mathcal{B}(\theta) \) are continuously differentiable. Since the closure of \( D_R \) is compact, it follows from Coddington (1961), Theorem 1 of Ch. 6, that \( \mathcal{A}(\theta) \) and \( \mathcal{B}(\theta) \) are Lipschitz continuous on \( D_R \). The rest of assumptions A are immediate given the gain sequence \( 1/t \).
We now turn to the projection facility. Because $\theta^*$ is Lyapunov stable there exists an associated Lyapunov function $U : D \to \mathbb{R}_+$. (See Theorem 11.1 of Krasovskii (1963) or Proposition 5.9, p. 98, of Evans and Honkapohja (2001).) For $c > 0$, define the notation

$$K(c) = \{ \theta \in D : U(\theta) \leq c \}.$$  

Pick $c_1 > 0$ such that $K(c_1) \subset D_R$, and let $D_1 = \text{int}(K(c_1))$. Pick $c_2 < c_1$, and let $D_2 = K(c_2)$. Let $\hat{\theta}_t \in D_2$. Define a new recursive algorithm for $\theta_t$ as follows:

$$\theta_t = \begin{cases} 
\hat{\theta}_t = \theta_{t-1} + \frac{1}{t} \mathcal{H}(\theta_{t-1}, X_t) & \text{if } \hat{\theta}_t \in D_1 \\
\theta_t & \text{if } \hat{\theta}_t \notin D_1
\end{cases}.$$  

With these definitions, Theorem 4 of Ljung applies and shows that $\theta_t \to \theta^*$ almost surely.

We remark that if $\varepsilon_t$ does not have compact support but has finite absolute moments then it is possible to use Ljung’s assumptions B or the results presented on p. 123 – 125 of Evans and Honkapohja (2001) and Corollary 6.8 on page 136.

**Discussion of exogenous states assumption that $m = n_2$.**

Recall the LQ-problem\(^{28}\)

$$V(x_0) = \max_{x_{t+1}} - \sum x_t'Rx_t + u_t'Qu_t$$  

(61)

$$x_{t+1} = Ax_t + Bu_t.$$  

(62)

As usual, we assume that $R, Q, A, \text{ and } B$ satisfy LQ.1 – LQ.3. Also, as in the main text, there are $n_1$ exogenous states, $n_2$ endogenous states, and $m$ controls, where we write $x_{1t}, x_{2t}$ as the exogenous and endogenous state, respectively.\(^{29}\) It will be convenient to use the “direct sum” notation: if the matrix $X_1$ is $p \times p$ and the matrix $X_2$ is $q \times q$ then

$$X_1 \oplus X_2 = \begin{pmatrix} X_1 & 0 \\
0 & X_2\end{pmatrix}.$$  

Because there are $n_1$ exogenous states it follows that $A = A_{11} \oplus 0_{n_2}$ and

$$B = \begin{pmatrix} 0_{n_1 \times m} \\
B_2\end{pmatrix},$$  

where $0_{n_1 \times n_2}$ is a $n_1 \times n_2$ matrix of zeros, and $0_{*}$ is a square matrix of zeros with dimension $* \times *$.

\(^{28}\) We consider the non-stochastic case for simplicity. The stochastic case obtains via an analogous argument.

\(^{29}\) We may assume that $x_{20} = 0$.  

45
We show that without loss of generality, we may assume \( n_2 = m \). Because of the assumptions on \( A \) and \( B \), a sequence of controls \( \{u_t\} \) uniquely identifies the sequence of corresponding endogenous states: \( x_{2t+1}(u_t) = B_2 u_t \). Our goal define a new sequence of states \( z_{t+1}(u_t) \in \mathbb{R}^m \), and a modified collection matrices \( \tilde{R}, \tilde{Q}, \tilde{A} \) and \( \tilde{B} \) so that

1. The modified collection satisfied LQ.1 – LQ.3.

2. Given any control path \( u_t \), if \( x_{2t} = B_2 u_t \) and \( z_t = \tilde{B}_2 u_t \) then

\[
x'_t R x_t + u'_t Q u_t = \tilde{x}'_t \tilde{R} \tilde{x}_t + u'_t \tilde{Q} u_t,
\]

where \( \tilde{x}'_t = (x'_{1t}, z'_t) \).

Now suppose that items 1 and 2 have been proven. Then the modified problem

\[
\tilde{V}(\tilde{x}_0) = \max \left\{ \sum (\tilde{x}'_t \tilde{R} \tilde{x}_t + u'_t \tilde{Q} u_t) \middle| \tilde{x}_{t+1} = \tilde{A} \tilde{x}_t + \tilde{B} u_t \right\}
\]

is equivalent to (61) in the following sense: \( \{u^*_t\} \) unique solution (64) if and only if it is the unique solution to (61), and \( \tilde{V}(\tilde{x}_0) = V(x_0) \).\(^{30}\)

Establishing items 1 and 2 involves constructing the modified matrices, verifying that they satisfy LQ.1 – LQ.3, and demonstrating that (63) holds. There are two cases.

Case 1: \( n_2 < m \). The intuition for this case is straightforward: we simply expand the state vector to include controls. More specifically, let \( N = m - n_2 \) and define

\[
\tilde{A} = A_{11} \oplus 0_{n_2} \oplus 0_N \text{ and }
\]

\[
\tilde{B} = \begin{pmatrix}
0_{n_1 \times m} \\
B_2 \\
0_{N \times m - N} \quad I_N
\end{pmatrix}.
\]

Next, pick \( \varepsilon > 0 \) small enough so that \( \tilde{Q} = Q - 0_{m-N} \oplus \varepsilon I_N \) is positive definite, and let

\[
\tilde{R} = R \oplus 0_N + 0_n \oplus \varepsilon I_N.
\]

Then \( \tilde{R} \) is positive semi-definite. Also, since \( A_{11} \) is stable, the pair \( (\tilde{A}, \tilde{B}) \) is trivially stabilizable. To assess observability, note that the eigenvectors of \( \tilde{A} \) take the form

\(^{30}\)Here, \( \tilde{x}_0 = (x'_{10}, 0) \).
Lemma 5. Let $\tilde{P}_{n_{1}n_{2}}$ and $(0', y_2')$, with $y_1 \in \mathbb{R}^{n_{1}}$ and $y_2 \in \mathbb{R}^{n_{2}}$. If $\tilde{R}(y_1', 0') = 0$ then $y_1 = 0$ by the observability of $(A, R)$. The invariant subspace comprised of eigenvectors of the form $(0', y_2')$ is spanned by the coordinate vectors $e_i$ for $i > n_1$. If $n_1 + 1 \leq i \leq n_1 + n_2$ then $\tilde{R}e_i = \epsilon e_i \neq 0$, where the “not equals” follows from the observability of $(A, R)$. If $i > n_2$ then $\tilde{R}e_i = \epsilon e_i \neq 0$. Observability of $(\tilde{A}, \tilde{R})$ follows. Finally, equation (63) follows by construction.

Case 2: $n_2 > m$. Let $r = \text{rank}(B)$ and $W \subset \mathbb{R}^{n_{2}}$ be the image $B$. Choose an orthonormal basis $\{w_1, \ldots, w_r\}$ for $W$ and let $S$ be the $r \times n_{2}$ matrix with $i^{th}$-row corresponding to the coordinates of $w_i \in \mathbb{R}^{n_{2}}$. We are now ready to identify the modified matrices. Let $\tilde{R} = (I_{n_1} \oplus S)R(I_{n_1} \oplus S')$, $\tilde{B}_2 = SB_2$ and $\tilde{A} = A_{11} \oplus 0_r$; the matrix $Q$ is unchanged. Given a control sequence $u$, the modified endogenous states are $z_t = \tilde{B}_2u_t \in \mathbb{R}^{r}$. As above, $\tilde{R}$ is positive semi-definite, and $(\tilde{A}, \tilde{B})$ is trivially stabilizable. To establish observability we again need only consider vectors of the form $(0, y_2')$, with $y_2 \in \mathbb{R}^{r}$. But $\tilde{R}(0, y_2') = 0$ implies $R(0, x_2') = 0$ for some $x_2 \in W \subset \mathbb{R}^{n_{2}}$. But since $(0, y_2')$ is an eigenvector of $A$ it follows that $x_2 = 0$ by the observability of $(A, R)$. And, finally, equation (63) again follows by construction. We have now reduced the dimension of the endogenous state space to $r \leq m$. If $r < m$, we may expand the state space as in Case 1 above.

Proof of Theorem 5. First, notice that since $B' H = B'_2 H_2$, it follows that

$$
(2Q - \beta B' H B)^{-1}(\beta B' H A - 2W') = (2Q - \beta B'_2 H_2 B)^{-1}(\beta B'_2 H_2 A - 2W'),
$$

whence

$$
\hat{F}(H_2) = F(H, A, B). \quad (66)
$$

In particular $\hat{F}(\hat{H}^*) = F^*$, proving part one.

Next, we use (22) to compute the last $m = n_{2}$ rows of $T^{SP}(H)$:

$$
T^{SP}(H)_{2} = (-2R - 2W F(H, A, B) + \beta A' H (A + BF(H, A, B)))_{2}
= -2R_{2} - 2W_{2} \hat{F}(H_{2}) + \beta A'_{22} H_2 \left( A + B_2 \hat{F}(H_2) \right) = \hat{T}^{SP}(H_2). \quad (67)
$$

where the second equality comes from (66) and the block-matrix forms of $A$ and $B$, and we recall that the notation $(\ast)_{2}$ identifies the last $n_{2}$ rows of the matrix $(\ast)$.

Note

$$
T^{SP}(H) = \begin{pmatrix} T^{SP_1}(H) \\ T^{SP_2}(H) \end{pmatrix},
$$

so that, by (67), $T^{SP_2}(H) = \hat{T}^{SP}(H_2)$. It follows that

$$
\hat{H}^* = H^*_2 = T^{SP}(H^*) = \tilde{T}^{SP}(H^*) = \hat{T}^{SP}(\hat{H}^*),
$$

which establishes part 2.
To assess stability, first note that by (67),

\[ \text{vec}(\hat{H}_2) = \hat{T}_v^{SP}(\text{vec}(H_2)) - \text{vec}(H_2). \]  

(68)

Now let \( z = (z_1, z_2) \equiv (\text{vec}(H_1)', \text{vec}(H_2)')' \), let \( S \) be the unitary matrix so that \( z = S \cdot \text{vec}(H) \), and consider the permuted system

\[ \hat{z} = S \cdot T_v^{SP}(S' \cdot z) - z \equiv G(z) - z, \]

where the equivalence \( \equiv \) defines the temporary notation \( G \). If \( G = (G_1', G_2')' \) follows that \( \hat{z}_2 = G_2(z) - z_2 \). Since \( z_2 = \text{vec}(H_2) \), we conclude by (68) that \( G_2(z) = \hat{T}_v^{SP}(z_2) \).

Differentiating \( G \), we compute

\[
D_z G(z^*) = \begin{pmatrix}
D_{z_1}G(z^*) & D_{z_2}G(z^*) \\
0 & D_{z_2}\hat{T}_v^{SP}(z_2^*)
\end{pmatrix},
\]

(69)

where \( z^* = S \cdot \text{vec}(H^*) \). It follows that

\[
eig \circ D\hat{T}_v^{SP}(\text{vec}(\hat{H}^*)) = \eig \circ D_{z_2}\hat{T}_v^{SP}(z_2^*) \\
\subseteq \eig \circ D_z G(z^*) \\
= \eig \circ (S \cdot DT_v^{SP}(\text{vec}(H^*)) \cdot S') \\
= \eig \circ DT_v^{SP}(\text{vec}(H^*)).
\]

Because \( S \) is unitary, it follows that \( S' = S^{-1} \). Since the eigenvalues of similar matrices are equal, part three is established by appealing to Theorem 3.

**Proof of Theorem 6.** We begin by noting that we can write \( q'z = -x'\mathcal{M}(q)x \) for a uniquely-defined symmetric matrix \( \mathcal{M}(q) \). Clearly the map \( \mathcal{M} \) is an isomorphism, i.e. linear and invertible. Next we define

\[
\delta(q) = -\frac{\beta}{1 - \beta} \text{tr} (\sigma_q^2 \mathcal{M}(q)CC') \quad \text{and the } n \times n \text{ matrix } \Delta(q) = \delta(q) \oplus 0,
\]

where 0 is the \((n-1) \times (n-1)\) zero matrix. Finally we define \( \hat{\mathcal{M}}(q) = \mathcal{M}(q) + \Delta(q) \).

Let \( N = n(n+1)/2 \), and notice that \( \hat{\mathcal{M}}(q) : \mathbb{R}^N \to S(n) \subset \mathbb{R}^{n \times n} \), where \( S(n) \) denotes the \( N \)-dimensional vector space of \( n \times n \) symmetric matrices. We need to show that \( \hat{\mathcal{M}} \) is invertible. Since \( \Delta(q) \) is linear in \( q \), it follows that \( \hat{\mathcal{M}} \) is linear in \( q \); thus invertibility is implied if \( \hat{\mathcal{M}} \) is injective. Suppose \( \mathcal{M}(q) = 0 \). Since \( \mathcal{M} \) is an isomorphism and all except the \((1,1)\) entry of \( \mathcal{M} \) and \( \hat{\mathcal{M}} \) are equal, it follows that \( q_i = 0 \) for \( i \neq 1 \). Also, since \( C_{1j} = 0 \) for \( 1 \leq j \leq k \), we conclude that \( \partial \delta(q)/\partial q_1 = 0 \). This implies that \( \delta(q) = \delta(0) = 0 \), so that \( \mathcal{M}(q) = \hat{\mathcal{M}}(q) \); since \( \mathcal{M} \) is injective, we conclude \( q = 0 \), and the injectivity of \( \hat{\mathcal{M}} \) is established.

We next claim that

\[
T^{VF}(q) = \mathcal{M}^{-1} \left( T(\hat{\mathcal{M}}(q)) - \Delta(q) \right).
\]

(70)
Observe that for choice \( \hat{u}(x) \) we have that 
\[-x' \mathcal{M}(T^{VF}(q)) x = T^{VF}(q)' z(x) = \]

\[
\begin{align*}
&= r(x, \hat{u}(x)) - \beta E (A x + B \hat{u}(x) + C \varepsilon)' \mathcal{M}(q) (A x + B \hat{u}(x) + C \varepsilon) \\
&= r(x, \hat{u}(x)) - \beta E (A x + B \hat{u}(x) + C \varepsilon)' \left( \hat{\mathcal{M}}(q) - \Delta(q) \right) (A x + B \hat{u}(x) + C \varepsilon) \\
&= r(x, \hat{u}(x)) - \beta (A x + B \hat{u}(x))' \hat{\mathcal{M}}(q) (A x + B \hat{u}(x)) \\
&\quad + \beta (A x + B \hat{u}(x))' \Delta(q) (A x + B \hat{u}(x)) - \beta E \varepsilon' C' \left( \hat{\mathcal{M}}(q) - \Delta(q) \right) C \varepsilon \\
&= -x' T \left( \hat{\mathcal{M}}(q) \right) x + \beta \delta(q) - \beta \text{tr} \left( \sigma_\varepsilon^2 \hat{\mathcal{M}}(q) C C' \right) \\
&= -x' T \left( \hat{\mathcal{M}}(q) \right) x + \beta \delta(q) - \beta \text{tr} \left( \sigma_\varepsilon^2 \mathcal{M}(q) C C' \right) \\
&= -x' T \left( \hat{\mathcal{M}}(q) \right) x + \delta(q) = -x' \left( T \left( \hat{\mathcal{M}}(q) \right) - \Delta(q) \right) x.
\end{align*}
\]

Here the fifth line follows from (i) the structure of \( \Delta(q) \) because \( A x + B \hat{u}(x) \) has one as the first entry, which implies \( (A x + B \hat{u}(x))' \Delta(q) (A x + B \hat{u}(x)) = \delta(q) \), and (ii) follows because the first row of \( C \) has zero entries, which implies \( \Delta(q) C C' = 0 \). Thus 
\[ \mathcal{M}(T^{VF}(q)) = T \left( \hat{\mathcal{M}}(q) \right) - \Delta(q) \], and our claim is proved.

Now let \( P^* \) be the solution to the Riccati equation and define \( q^* = \hat{\mathcal{M}}^{-1}(P^*) \). By construction,
\[ q^* \cdot z = V(x) = -x' P^* x - \frac{\beta}{1 - \beta} \text{tr} \left( \sigma_\varepsilon^2 P^* C C' \right). \]

Also, by our claim,
\[ T^{VF}(q^*) = \mathcal{M}^{-1} \left( T(\hat{\mathcal{M}}(q^*)) - \Delta(q^*) \right) = \mathcal{M}^{-1} \left( \hat{\mathcal{M}}(q^*) - \Delta(q^*) \right) = \mathcal{M}^{-1}(\mathcal{M}(q^*)) = q^*, \]
so that \( q^* \) is indeed a rest point of the ordinary differential equation (ode) \( dq/d\tau = T^{VF}(q) - q \).

We now turn to stability analysis. First, we establish
\[ \text{eig} \circ D(T^{VF}) = \text{eig} \circ D \left( \mathcal{M}^{-1} \circ T \circ \hat{\mathcal{M}} \right). \quad (71) \]

This will take several steps.

**Step 1.** Recall from above that \( \partial \delta(q)/\partial q_1 = 0 \). It follows that
\[ D \mathcal{M}^{-1} (\Delta(q^*)) = \left( \begin{array}{cc} 0_{1 \times 1} & Z(1) \\ 0 & 0 \end{array} \right), \quad (72) \]
for appropriate \( 1 \times N - 1 \) matrix \( Z(1) \).
Step 2. Our goal here is to show that

$$D \left( \mathcal{M}^{-1} \circ T \circ \hat{\mathcal{M}} \right) = \begin{pmatrix} Z(2) & Z(3) \\ 0 & Z(4) \end{pmatrix} \tag{73}$$

where $Z(2)$ is a $1 \times 1$ matrix, and $Z(3)$ and $Z(4)$ are conformable. It suffices to show that $\frac{\partial}{\partial q_i} \left( \mathcal{M}^{-1} \circ T \circ \hat{\mathcal{M}} \right)_j = 0$, for $2 \leq j \leq N$, which itself would follow by showing $\frac{\partial}{\partial q_i} \left( \text{vec} \circ T \circ \hat{\mathcal{M}} \right)_j = 0$, for $2 \leq j \leq N$.\(^{31}\) To show this, notice that

$$\text{vec} \circ T \circ \hat{\mathcal{M}} = \left( \text{vec} \circ T \circ \text{vec}^{-1} \right) \circ \left( \text{vec} \circ \hat{\mathcal{M}} \circ \text{vec}^{-1} \right) = T_v \circ \hat{\mathcal{M}}_v,$$

where the operator $\text{vec}^{-1}$, applied just before (i.e. to the right of) $\hat{\mathcal{M}}$, is the identity map. Thus we study the first column of $D \left( T_v \circ \hat{\mathcal{M}}_v \right) (q^*)$.

Now, recall that

$$D(T_v)(P) = (\Omega(P)' \otimes \Omega(P)') \equiv \Gamma^T(P),$$

where

$$\Omega(P) = \beta^{1/2} A - \beta^{1/2} B(Q + \beta B'PB)^{-1}(\beta B'PA + W').$$

Since $B_{1j} = 0$ for $1 \leq j \leq m$ and $A_{1j} = 0$ for $2 \leq j \leq n$, it follows that $\Omega(P)_{1j} = 0$ for $2 \leq j \leq n$. Using the definition of the tensor product, we conclude that $\Gamma^T(P)_{j1} = 0$ for $2 \leq j \leq n^2$.

Next, notice that

$$D(\hat{\mathcal{M}}_v)_{j1} = D(\mathcal{M}_v)_{j1} + D(\text{vec} \circ \Delta \circ \text{vec}^{-1})_{j1} = 0,$$

for $2 \leq j \leq n^2$. Letting $D(\hat{\mathcal{M}}_v)(q) = \Gamma^{\hat{\mathcal{M}}}(q)$, and applying the chain rule, it follows that

$$\left( D \left( T_v \circ \hat{\mathcal{M}}_v \right) (q^*) \right)_{j1} = \left( \Gamma^T(P^*) \cdot \Gamma^{\hat{\mathcal{M}}}(q^*) \right)_{j1} = 0$$

for $2 \leq j \leq n^2$, which completes step 2.

Step 3. We now use steps 1 and 2 to demonstrate equation (71). Since

$$\left( \mathcal{M}^{-1} \circ T \circ \mathcal{M} \right)_j = \left( \hat{\mathcal{M}} \circ T \circ \hat{\mathcal{M}} \right)_j$$

for $2 \leq j \leq N$, it follows that

$$D \left( \hat{\mathcal{M}}^{-1} \circ T \circ \hat{\mathcal{M}} \right) = \begin{pmatrix} \hat{Z}(2) & \hat{Z}(3) \\ 0 & Z(4) \end{pmatrix}, \tag{74}$$

\(^{31}\)A point on notation: if $f : \mathbb{R}^m \to \mathbb{R}^n$ then $\partial / \partial x_i(f)_j$ is the partial derivative of the $j$-th coordinate of $f$, taken with respect to $x_i$, with the point of evaluation implied.
with $Z(4)$ as in (73), and $\hat{Z}(2)$ and $\hat{Z}(3)$ conformable. By steps 1 and 2, then, it suffices to show that $Z(2) = \hat{Z}(2)$. Note that we may write
\[
\hat{\mathcal{M}}^{-1}(P) = \mathcal{M}^{-1}(P) + \hat{\Delta}(P),
\]
where $\hat{\Delta}(P) = \hat{\delta}(P) \oplus 0$, and $\frac{\partial}{\partial P}(\hat{\delta}) = 0$. It follows that if $D(\hat{\Delta}_v)(vec(P)) = \Gamma^\Delta(vec(P))$, then $\Gamma^\Delta(vec(P))_{j1} = 0$ for $1 \leq j \leq N$. Since $\hat{\Delta} \circ T \circ \hat{\mathcal{M}} = \hat{\Delta}_v \circ T_v \circ \hat{\mathcal{M}}_v$, we compute
\[
\begin{align*}
\left( D\left( \hat{\Delta} \circ T \circ \hat{\mathcal{M}} \right) (q^*) \right)_{j1} &= \left( D\left( \hat{\Delta}_v \circ T_v \circ \hat{\mathcal{M}}_v \right) (q^*) \right)_{j1} \\
&= \left( \Gamma^\Delta(vec(P^*)) \cdot \Gamma^T(vec(P^*)) \cdot \Gamma^\hat{\mathcal{M}}(q^*) \right)_{j1} = 0
\end{align*}
\]
for $1 \leq j \leq N$, and conclude
\[
\frac{\partial}{\partial q_1} \left( \hat{\Delta} \circ T \circ \hat{\mathcal{M}} \right) = 0.
\]
Thus
\[
\hat{Z}(2) = \frac{\partial}{\partial q_1} \left( \hat{\mathcal{M}}^{-1} \circ T \circ \hat{\mathcal{M}} \right) = \frac{\partial}{\partial q_1} \left( \mathcal{M}^{-1} \circ T \circ \hat{\mathcal{M}} + \hat{\Delta} \circ T \circ \hat{\mathcal{M}} \right) = \frac{\partial}{\partial q_1} \left( \mathcal{M}^{-1} \circ T \circ \hat{\mathcal{M}} \right) = Z(2),
\]
and step 3 is complete.

By equation (71), stability may be assessed by studying
\[
eig \circ D\left( \hat{\mathcal{M}}^{-1} \circ T \circ \hat{\mathcal{M}} \right) (q^*).
\]
Since $\hat{\mathcal{M}}^{-1} \circ T \circ \hat{\mathcal{M}} = \hat{\mathcal{M}}_v^{-1} \circ T_v \circ \hat{\mathcal{M}}_v$, we may apply the chain rule to get
\[
D \left( \hat{\mathcal{M}}^{-1} \circ T \circ \hat{\mathcal{M}} \right) (q^*) = D \left( \hat{\mathcal{M}}_v^{-1} \right) (vec(P^*)) \cdot D(T_v)(P^*) \cdot D(\hat{\mathcal{M}}_v)(q^*).
\]
Since
\[
D \left( \hat{\mathcal{M}}_v^{-1} \right) (vec(P^*)) = \left( D(\hat{\mathcal{M}}_v)(q^*) \right)^{-1},
\]
we conclude that the eigenvalues of $DT^{VF}(q^*)$ are equal to the eigenvalues of $D(T_v)(vec(P^*))$, and the result follows from Lemma 2. ■

**Proof of Theorem 7.** We begin by computing matrix differentials. At arbitrary (appropriate) $F$, we have that $dT^{EL} = -(d\Phi \cdot \Psi + \Phi \cdot d\Psi)$, and
\[
\begin{align*}
d\Phi &= -((Q + \beta B'(R + WF)B)^{-1}\beta B'W) \cdot dF \cdot (B(Q + \beta B'(R + WF)B)^{-1}) \\
d\Psi &= (\beta B'W) \cdot dF \cdot (A).
\end{align*}
\]
Evaluated at $F^*$, we obtain
\[ dT^{EL} = (\beta \Phi(P^*)B'W) \cdot dF \cdot (B\Phi(P^*)\Psi(P^*) - A). \]

Applying the vec operator to each side, we obtain
\[ D \left( T_v^{EL} \right) (F^*) = -\Omega(P^*)' \otimes \beta^{1/2}\Phi(P^*)B'W, \]
where we recall that
\[
\Omega(P^*) = \beta^{1/2}A - \beta^{1/2}B(Q + \beta B'P^*B)^{-1}(\beta B'P^*A + W')
\]
\[ = \beta^{1/2}(A - \Phi(P^*)\Psi(P^*)). \]

In the proof of Lemma 2 it is shown that $\text{eig} \circ \Omega(P)$ have modulus less than one; thus the proof will be complete if we can show that the eigenvalues of $\Phi(P^*)B'W$ are inside the unit circle. This requires three steps.

**Step 1.** We show that the eigenvalues of $B\Phi(P^*)\Psi(P^*)$ are inside the unit circle. To this end, we establish
\[
\text{eig} \circ \Omega(P) = \beta^{1/2}(\text{eig} \circ A \cup \text{eig} \circ (-B\Phi(P^*)\Psi(P^))).
\]
To see this let $y = -B\Phi(P^*)\Psi(P^*)$ and notice that since the first $n_1$ rows of $B$ are zeros we have
\[
y = \begin{pmatrix}
0 \\
y_{21} \\
y_{22}
\end{pmatrix}.
\]
Because $A = A_{11} \oplus 0$, we conclude that
\[
A - \Phi(P^*)\Psi(P^*) = \begin{pmatrix}
A_{11} \\
y_{21} \\
y_{22}
\end{pmatrix},
\]
and step 1 is complete by Lemma 2.

**Step 2.** Now we show that
\[
\text{eig} \circ B\Phi(P^*)\Psi(P^*) = \text{eig} \circ B\Phi(P^*)W'.
\]
Let $yB\Phi(P^*)B'P^*A$ and $y = B\Phi(P^*)W'$, so that $B\Phi(P^*)\Psi(P^*) = y + z$. The block-diagonal structure of $B$ implies
\[
y = \begin{pmatrix}
0 \\
y_{21} \\
y_{22}
\end{pmatrix} \quad \text{and} \quad z = \begin{pmatrix}
0 \\
z_{21} \\
z_{22}
\end{pmatrix},
\]
and step 2 is follows.

**Step 3.** Finally, we show that
\[
\text{eig} \circ \Phi(P^*)B'W \subset \text{eig} \circ B\Phi(P^*)W'.
\]
Since eigenvalues are preserved under transposition, it suffices to show that
\[ \text{eig} \circ W' B \Phi(P^*) \subset \text{eig} \circ B \Phi(P^*) W'. \]

To this end, let \( y = B \Phi(P^*) \), and notice \( y' = \begin{pmatrix} 0 & y_2' \end{pmatrix} \). Writing the \( n \times m \) matrix \( W \) as \( W' = \begin{pmatrix} W_1' & W_2' \end{pmatrix} \), we compute \( W'y = W_2' y_2 \) and
\[
y W' = \begin{pmatrix} 0 & 0 \\ y_2 W_1' & y_2 W_2' \end{pmatrix}.
\]

Since \( y_2 \) and \( W_2 \) are \( m \times m \) matrices, it follows that \( \text{eig} \circ y_2 W_2' = \text{eig} \circ W_2' y_2 \), and step 3 is complete.

Finally, the proof is completed by observing
\[
\text{eig} \circ \Phi(P^*) B' W \subset \text{eig} \circ B \Phi(P^*) W' = \text{eig} \circ B \Phi(P^*) \Psi(P^*) \subset S^1,
\]
which follow by applying the steps in reverse order. ■

**Generalization of Theorem 1.** Theorem 1 is a standard formulation using assumption LQ.3, which requires that the objective observe the eigenvectors of the state-variable transition matrix: \( Ay = \lambda y \) implies \( Ry \neq 0 \). This assumption, while standard, is somewhat stronger than necessary: only observability in this sense of the endogenous states is required. Furthermore, application of our results to standard setups is facilitated by this weaker assumption, which we now formalize as LQ.3’ below.

Recall the transformed problem
\[
\max \quad -E_0 \sum \left( \dot{x}_t^i R \dot{x}_t^i + u_t^i Q u_t^i \right) \\
\text{s.t.} \quad \dot{x}_{t+1}^i = \dot{A} x_t^i + \dot{B} u_t + \beta_{1+}^i C \varepsilon_{t+1},
\]
where, to identify exogenous and endogenous states, we now impose
\[
\dot{A} = \begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix}, \quad \text{and} \quad B = \begin{pmatrix} 0 \\ B_2 \end{pmatrix}.
\]

As before, we assume, without loss of generality, that there are \( n_1 \) exogenous states, and that \( n = n_1 + n_2 \). The associated Riccati equation is given by
\[
P = \dot{R} + \dot{A}' P \dot{A} - \dot{A}' P \dot{B} \left( Q + \dot{B}' P \dot{B} \right)^{-1} \dot{B}' P \dot{A}.
\]

We modify LQ.3 as follows:

LQ.3’: Recall \( \dot{R} = \dot{D} \dot{D}' \).

53
a. If $\lambda \in e^{ig}(A_{11})$ then $|\lambda| < 1$;

b. If $\dot{A}y = \lambda y$ and $\dot{D}'y = 0$ then $y = (y',0)'$.

First notice that LQ.3' implies detectability of $(\hat{A}, \hat{D})$: if $y = (y',0)'$ then

$$\dot{A}y = \begin{pmatrix} A_{11}y_1 \\ 0 \end{pmatrix} \text{ and } \dot{A}y = \lambda \begin{pmatrix} y_1 \\ 0 \end{pmatrix}.$$  

Thus $\lambda \in e^{ig}(A_{11})$, so that $|\lambda| < 1$. It follows that our workhorse result, Lemma 2, which requires detectability, continues to hold under LQ.1, LQ.2, and LQ.3'. We have the following result:

**Theorem 1'** Under assumptions LQ.1, LQ.2, and LQ.3', the Riccati equation (76) has a unique positive semi-definite solution, $P^*$. Also, there is a unique sequence of controls solving (7), and they are given by $u_t = \hat{F}\hat{x}_t$, where

$$F = -\left(Q + \hat{B}'P^*\hat{B}\right)^{-1}\hat{B}'P^*\hat{A}. \quad (77)$$

**Proof.** We proceed in several steps.

Step 1: Existence of $P^*$. The existence of a unique positive semi-definite solution $P^*$ to the Riccati equation (76) follows from Lancaster and Rodman (1995), Theorem 13.5.2.

Step 2: Constructing $\tilde{R}$. The idea is to modify the objective in a manner that allows Theorem 1 to apply, but so that the control choice is not altered. To this end, let $r \leq n$ be the rank of $\hat{R}$, and let $N = \min\{n - r, n_2\}$. Write $\hat{R} = \hat{D}\hat{D}'$ with $\dim(\hat{D}) = n \times r$, $\text{rank}(\hat{D}) = r$ and define

$$\tilde{D} = \begin{pmatrix} I_N & \hat{D}_1 \\ 0 & \hat{D}_2 \end{pmatrix},$$

where $\hat{D} = (\hat{D}_1', \hat{D}_2')'$, with $\dim(\hat{D}_1) = n_1 \times r$ and $\dim(\hat{D}_2) = n_2 \times r$. We define $\tilde{R} = \tilde{D}\tilde{D}'$, and note that $\tilde{R}$ is positive semi-definite. This completes step 2.

Step 3: That $(\hat{A}, \tilde{R})$ is observable. There are two cases to consider.

1. $N = n_2$. Let $\dot{A}y = \lambda y$ and $\dot{D}'y = 0$. Since $\tilde{D}$ has full rank, it follows that $\tilde{D}'y = 0$. Computing, we obtain

$$\tilde{D}'y = 0 \iff \begin{pmatrix} I_N & 0 \\ D_1' & D_2' \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ D_1'y_1 + D_2'y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (78)$$

It is immediate from (78) that $\tilde{D}'y = 0 \implies y_1 = 0$; and by LQ.3', if $\hat{D}_1'y_1 + \hat{D}_2'y_2 = 0$ then $y_2 = 0$. 

54
2. $N = n - r$. In this case,

$$\text{rank}(\tilde{R}) = \text{rank}(\tilde{D}) = \text{rank}(\hat{D}) + \text{rank}(I_{n-r}) = n,$$

so that $\tilde{R}$ is invertible, and observability follows.

**Step 4:** The modified LQ-problem. We now consider the modified LQ-problem obtained by replacing the loss in (75) with $E_0 \left( \dot{x}_t e^T + u_t^T Q u_t \right)$. This modified problem satisfies the LQ.1 – LQ.3, and thus has a unique solution determined by feedback matrix $\tilde{F}$, where $\tilde{P}$ is the unique positive definite solution to the Riccati equation\(^{32}\)

$$\tilde{P} = \tilde{R} + \hat{A}' \tilde{P} \hat{A} - \hat{A}' \tilde{P} \hat{B} \left( Q + \hat{B}' \tilde{P} \hat{B} \right)^{-1} \hat{B}' \tilde{P} \hat{A},$$

(79)

and

$$\tilde{F} = - \left( Q + \hat{B}' \tilde{P} \hat{B} \right)^{-1} \hat{B}' \tilde{P} \hat{A}.$$  

Observing that $\tilde{R} = \tilde{R} + I_N \oplus 0$, we may compute

$$E_0 \left( \dot{x}_t \tilde{R} \dot{x}_t + u_t^T Q u_t \right) = E_0 \left( \dot{x}_t \left( \tilde{R} + I_N \oplus 0 \right) \dot{x}_t + u_t^T Q u_t \right)$$

$$= E_0 \left( \dot{x}_t \tilde{R} \dot{x}_t \right) + E_0 \left( \dot{x}_t \left( I_N \oplus 0_4 \right) \dot{x}_t \right) + E_0 \left( u_t^T Q u_t \right),$$

(80)

where $0_4$ has no dimension if $N = n_2$ and is conformable otherwise. Notice that the middle term of (80) is independent of the control. Thus

$$E_0 \left( \dot{x}_t \tilde{R} \dot{x}_t + u_t^T Q u_t \right) = E_0 \left( \dot{x}_t \tilde{R} \dot{x}_t + u_t^T Q u_t \right) + \text{constant.}$$

It follows that the feedback rule $u = \tilde{F} \dot{x}$ also solves the original LQ-problem (75).

**Step 5:** $F = \tilde{F}$. By step 1, the Riccati equation (76) has a unique positive semi-definite solution $P^*$, from which we can form the feedback matrix $F$ using (77). By step 4, the LQ-problem (75) has a unique solution given by $u = \tilde{F} \dot{x}$. It remains to show that $F = \tilde{F}$. To this establish this, we first study $\tilde{P}$. We claim that $\tilde{P} = P^* + S \oplus 0$, where

$$S = \sum_{t \geq 0} (\hat{A}'_{11})^t (I_N \oplus 0_4) (\hat{A}_{11})^t,$$  

(81)

which is well-defined because the eigenvalues of $A_{11}$ are contracting. Note that $S$ is positive semi-definite and $\text{dim}(S) = N \times N$. Replacing $\tilde{P}$ with $P^* + S \oplus 0$ in (79), exploiting that $(S \oplus 0) B = 0$, and simplifying, we obtain

$$\hat{R} + I_N \oplus 0 + \hat{A}' P^* \hat{A} + \hat{A}' (S \oplus 0) \hat{A} - \hat{A}' P^* \hat{B} \left( Q + \hat{B}' P^* \hat{B} \right)^{-1} \hat{B}' P^* \hat{A}$$

$$= P^* + I_N \oplus 0 + \hat{A}' (S \oplus 0) \hat{A},$$

\(^{32}\) The solution is positive definite (instead of positive semi-definite) since $\tilde{R}$ is invertible.
where the equality obtains since \( P^* \) solves (76). The result follows by noting

\[
I_N \oplus 0 + \bar{A}'(S \oplus 0)\bar{A} = I_N \oplus 0 + \bar{A}' \left( \sum_{t \geq 0} (\bar{A}'_{11})^t (I_N \oplus 0_*) (\bar{A}_{11})^t \right) \oplus 0 \bar{A}
\]

\[
= I_N \oplus 0 + \left( \sum_{t \geq 0} (\bar{A}'_{11})^{t+1} (I_N \oplus 0_*) (\bar{A}_{11})^{t+1} \right) \oplus 0 = S \oplus 0.
\]

Finally, since \( \bar{P} = P^* + S \oplus 0 \) and since \( (S \oplus 0)B = 0 \), we have

\[
\bar{F} = - \left( Q + \bar{B}'(P^* + S \oplus 0)\bar{B} \right)^{-1} \bar{B}'(P^* + S \oplus 0)\bar{A} = - \left( Q + \bar{B}'P^*\bar{B} \right)^{-1} \bar{B}'P^* \bar{A} = F,
\]

and the proof is complete. 

**Remark:** Because LQ.3' implies the detectability of \((\bar{A}, \bar{D})\), it straightforward to see from their proofs that Theorems 3 - 7 hold in each case if assumption LQ.3 is replaced by LQ.3'. Below we refer to these modified theorems as Theorems 3' - 7' respectively.

**Proof of Proposition 8.** We reproduce the LQ problem (42) here for convenience:

\[
\begin{align*}
\max_{t \geq 0} & \quad -E \sum_{t \geq 0} \beta^t \left( (c_t - b^*)^2 + \phi s_{t-1}^2 \right) \\
\text{s.t.} & \quad s_{t+1} = A_1 s_t + A_2 s_{t-1} - c_t + \mu_{t+1}
\end{align*}
\]

To place in standard LQ-form (see (7)), we define the state as \( x_t = (1, s_t, s_{t-1})' \) and the control as \( u_t = c_t \). Note that the intercept is an exogenous state. The key matrices are given by:

\[
R = \begin{pmatrix}
(b^*)^2 & 0 & 0 \\
0 & \phi & 0 \\
0 & 0 & 0
\end{pmatrix}, \quad A = \begin{pmatrix}
1 & 0 & 0 \\
0 & A_1 & A_2 \\
0 & 1 & 0
\end{pmatrix},
\]

and \( W = (-b^*, 0, 0)', B = (0, -1, 0)' \), and \( Q = 1 \). The transformed matrices are \( \hat{R} = R - WW', \hat{A} = \beta^{\frac{1}{2}} (A - BW') \), and \( \hat{B} = \beta^{\frac{1}{2}} B \): thus

\[
\hat{R} = \begin{pmatrix}
0 & 0 & 0 \\
0 & \phi & 0 \\
0 & 0 & 0
\end{pmatrix}, \quad \hat{A} = \beta^{\frac{1}{2}} \begin{pmatrix}
1 & 0 & 0 \\
-b^* & A_1 & A_2 \\
0 & 1 & 0
\end{pmatrix}.
\]

We see immediately that LQ.1 is satisfied: \( \hat{R} \) is positive semi-definite and \( Q \) is positive definite. Next let \( K = (K_1, K_2, K_3) \) be any \( 1 \times 3 \) matrix. Then

\[
\hat{A} - \hat{B}K = \beta^{\frac{1}{2}} \begin{pmatrix}
1 & 0 & 0 \\
-b^* + K_1 & A_1 + K_2 & A_2 + K_3 \\
0 & 1 & 0
\end{pmatrix}.
\]

56
By choosing $K_2 = -A_1$ and $K_3 = -A_2$, we see that $(\hat{A}, \hat{B})$ is a stabilizable pair: thus LQ.2 is satisfied. Finally, to verify LQ.3' note that

$$\hat{D} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sqrt{\phi} & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and suppose that $\hat{A}y = \lambda y$ and $\hat{D}y = 0$ is satisfied for some $y$. $\hat{D}y = 0$ implies that the second component of $y$ is zero, i.e. $y_2 = 0$. Since the third component of of $\hat{A}y$ must then be zero, it then follows from $\hat{A}y = \lambda y$ that $y_3 = 0$ or $\lambda = 0$. Since $A_2 \neq 0$ implies $\lambda \neq 0$, we conclude that LQ.3' is satisfied. The proof is completed by application of Theorem 4'.

**Details of Robinson Crusoe Economy** The LQ set-up (82) does not directly impose non-negativity constraints that are present. We show that under suitable assumptions these constraints are never violated. Carefully specified as an economics problem, (82) should include the constraints

$$0 \leq c_t \leq A_1 s_t + A_2 s_{t-1} \text{ and } s_{t+1} \geq 0.$$  

(83)

In addition, because the assumption of “free disposal” is not incorporated in the LQ set-up, our solution must also obey $c_t \leq b^*$. We show these inequalities are satisfied by demonstrating that $s_t > 0$ and $c_t \in (0, b^*)$ under the assumptions specified in the text. Indeed, let $c$ and $s$ be the respective nonstochastic steady-state values of $c_t$ and $s_t$. The transition equation implies $c = \Theta s$, where $\Theta = A_1 + A_2 - 1$. Inserting this condition into the Euler equation, reproduced here for convenience,

$$c_t - \beta \phi E_t s_{t+1} = b^*(1 - \beta A_1 - \beta^2 A_2) + \beta A_1 E_t c_{t+1} + \beta^2 A_2 E_t c_{t+2},$$

(84)

and solving for $s$ yields

$$s = \frac{b^*(1 - \beta A_1 - \beta^2 A_2)}{\Theta(1 - \beta A_1 - \beta^2 A_2) - \beta \phi}.$$ 

If $\beta A_1 + \beta^2 A_2 > 1$ then $s > 0$ and $c \in (0, b^*)$. Provided suitable initial conditions hold and the support of $\mu_{t+1}$ is sufficiently small, it follows that $s_t > 0$ and $c_t \in (0, b^*)$.

**Euler Equation Learning in the Robinson Crusoe Economy** Turning to Euler equation learning, recall the PLM

$$c_t = a_3 + b_3 s_t + d_3 s_{t-1}.$$ 

Using this PLM, the following expectations may be computed:

$$\hat{E}_t c_{t+1} = a_3 + (b_3 A_1 + d_3) s_t + b_3 A_2 s_{t-1} - b_3 c_t$$

$$\hat{E}_t c_{t+2} = a_3 (1 - b_3) + ((b_3 (A_1 - b_3) + d_3) A_1 + b_3 (A_2 - d_3)) s_t + (b_3 (A_1 - b_3) + d_3) A_2 s_{t-1} - (b_3 (A_1 - b_3) + d_3) c_t.$$
Combining these expectations with (84) provides the following T-map:

\[ a_3 \rightarrow \frac{\psi + \beta A_1 a_3 + \beta^2 A_2 a_3(1 - b_3)}{1 + \beta \phi + \beta A_1 b_3 + \beta^2 A_2 (b_3(A_1 - b_3) + d_3)} \]

\[ b_3 \rightarrow \frac{\beta \phi A_1 + \beta A_1 (b_3 A_1 + d_3) + \beta^2 A_2 ((b_3(A_1 - b_3) + d_3)A_1 + b_3(A_2 - d_3))}{1 + \beta \phi + \beta A_1 b_3 + \beta^2 A_2 (b_3(A_1 - b_3) + d_3)} \]

\[ d_3 \rightarrow \frac{\beta \phi A_2 + \beta A_1 b_3 A_2 + \beta^2 A_2 (b_3(A_1 - b_3) + d_3)}{1 + \beta \phi + \beta A_1 b_3 + \beta^2 A_2 (b_3(A_1 - b_3) + d_3)} . \]
References


